KHBPBPCBAUTO
13383677

L-force *Controls*

Kommunikationshandbuch

PC-based Automation

Steuerungstechnik PROFIBUS

Inbetriebnahme & Konfiguration

Lenze

Inhalt

1 Über diese Dokumentation 5

1.1 Dokumenthistorie 7

1.2 Verwendete Konventionen 8

1.3 Verwendete Begriffe 9

1.4 Verwendete Hinweise 10

2 Sicherheitshinweise 11

3 Das System "PC-based Automation" 12

4 Das Lenze Steuerungssystem mit PROFIBUS 14

4.1 Kurzbeschreibung PROFIBUS 14

4.1.1 Aufbau des PROFIBUS-Systems 15

4.1.2 Prinzipielle Verdrahtung des PROFIBUS 17

4.1.3 Kombination mit anderen Bussystemen 18

4.1.4 Feldgeräte 18

4.2 PROFIBUS-Hardware für den Industrie-PC 19

5 Technische Daten 20

5.1 Technische Daten der Kommunikationskarte MC-PBM 20

5.2 Spezifikation des Buskabels 21

6 Inbetriebnahme des PROFIBUS 22

6.1 Übersicht der Inbetriebnahmeschritte 22

6.2 Detaillierte Inbetriebnahmeschritte 23

6.2.1 Bustopologie planen 23

6.2.2 Feldgeräte installieren 23

6.2.3 Projektordner anlegen 24

6.2.4 Gerätestammdatendateien (GSD) ablegen 24

6.2.5 Feldgeräte in Betrieb nehmen 25

6.2.6 Konfiguration im »PLC Designer« 26

7 CANopen mit PROFIBUS 37

8 Funktionsbibliotheken ... 38

8.1 Funktionsbibliothek BusDiag.lib ... 39
8.1.1 Funktionsbaustein DiagGetBusState ... 39
8.1.2 Funktionsbaustein DiagGetState ... 41

8.2 Funktionsbibliothek NetXPBInfo.lib ... 44
8.2.1 Struktur NETXGETPBINFOSTYP ... 44
8.2.2 Funktion NetXGetPBInfos ... 45

8.3 Funktionsbibliothek HilscherNetX.lib ... 46
8.3.1 Struktur CIFX_PACKET ... 46
8.3.2 Struktur CIFX_PACKET_HEADER ... 47
8.3.3 Funktion CIFXGetChannelHandle ... 48
8.3.4 Funktion CIFXPutPacket ... 48
8.3.5 Funktion CIFXGetPacket ... 49

8.4 Funktionsbibliothek SysLibDPV1Hilscher.lib ... 50
8.4.1 Struktur V1State ... 50
8.4.2 Funktionsbaustein DPV1_Read / DPV1_ReadEx ... 51
8.4.3 Funktionsbaustein DPV1_Write / DPV1_WriteEx ... 52
8.4.4 Telegramm-Beispiele für den PROFIdrive-Parameterdatenkanal (DP-V1) ... 53

9 Minimale Zykluszeit des PLC-Projektes bestimmen ... 57

9.1 Gesamtzugriffszeit auf die Peripheriegeräte berechnen ($T_{Korrektur}$) ... 57

9.2 Taskauslastung der Applikation ermitteln ($T_{Taskauslastung}$) ... 58
9.2.1 Anzeige der Systemauslastung im »PLC Designer« mit dem Task-Editor ... 58
9.2.2 Taskauslastung ermitteln ... 59

9.3 Minimale Zykluszeit berechnen ... 60

9.4 System optimieren ... 61

10 Diagnose ... 62

10.1 Diagnose im »PLC Designer« ... 62

10.2 Fehlermeldungen bei nicht vorhandener Kommunikationskarte MC-PBM ... 62

10.3 LED-Statusanzeigen ... 63

11 Parameter-Referenz ... 64

11.1 Parameter der Kommunikationskarte MC-PBM in Steckplatz 1 ... 65

11.2 Parameter der Kommunikationskarte MC-PBM in Steckplatz 2 ... 66

12 Index ... 67

1 Über diese Dokumentation

Diese Dokumentation ...

- ▶ enthält ausführliche Informationen zur Inbetriebnahme, Konfiguration und Diagnose des Bussystems PROFIBUS im Rahmen der Lenze Steuerungstechnik.
- ▶ ordnet sich in die Handbuchsammlung "PC-based Automation" ein. Diese besteht aus folgenden Dokumentationen:

Dokumentation	Thema
Systemhandbücher "PC-based Automation"	• Steuerungstechnik - *Systemaufbau & Konfiguration* • Visualisierung - *Systemaufbau & Komponenten*
Kommunikationshandbücher "PC-based Automation"	• Steuerungstechnik CANopen • Steuerungstechnik PROFIBUS • Steuerungstechnik EtherCAT
(Software-)Handbuch "PC-based Automation"	• Industrie-PC - *Parametrierung & Konfiguration*
Betriebsanleitung "Embedded Line Panel PC"	• EL x8xx - Einbau-Panel-PC mit TFT-Display
Betriebsanleitung "Command Station"	• CS x8xx - freistehendes Bedienterminal
Betriebsanleitung "Control Cabinet PC"	• CPC x8xx - Schaltschrank PC
Betriebsanleitung "HMI EL 100"	• EL 1xx - HMI mit Windows® CE
Weitere Softwarehandbücher	• »Global Drive Control« (»GDC«) – IPC als Gateway - *Parametrierung & Konfiguration* • »Engineer« • »PLC Designer« / »PLC Designer - SoftMotion« / »PLC Designer - CANopen für Laufzeitsysteme« • »VisiWinNET® Smart«

Weitere Technische Dokumentationen zu Lenze-Komponenten

Weitere Informationen zu Lenze-Komponenten, die in Verbindung mit "PC-based Automation" eingesetzt werden können, finden Sie in folgenden Dokumentationen:

Montieren & Verdrahten
MAs zu Inverter Drives 8400
MAs zu Servo Drives 9400
MA EPM-Txxx (I/O-System IP20)
MA EPM-Sxxx (I/O-System 1000)
MA 8200 vector
EMV-gerechte Verdrahtung 8200 vector
MAs zum Servo System ECS
MA Kommunikationskarte MC-CAN2
MA Kommunikationskarte MC-ETC
MA Kommunikationskarte MC-ETH
MA Kommunikationskarte MC-PBM
MA Kommunikationskarte MC-PBS
MA Kommunikationskarte MC-MPI
MAs zum Kommunikationsmodulen

Parametrieren, Konfigurieren, in Betrieb nehmen
SW Inverter Drive 8400 BaseLine / StateLine / HighLine / TopLine
SW Servo Drive 9400 HighLine / PLC
Inbetriebnahme-Leitfaden 9400 HighLine
SHB I/O-System IP20 (EPM-Txxx)
SHB I/O-System 1000 (EPM-Sxxx)
SHB 8200 vector
BAs zum zum Servo System ECS
KHBs zu Kommunikationsmodulen

Programmieren
SW Funktionsbibliothek 9400

Vernetzung aufbauen
KHBs zu Kommunikationsmodulen

Legende:

- Gedruckte Dokumentation
- Online-Hilfe/PDF

Verwendete Abkürzungen:

SHB	Systemhandbuch
BA	Betriebsanleitung
MA	Montageanleitung
SW	Software-Handbuch
KHB	Kommunikationshandbuch

Tipp!

Dokumentationen und Software-Updates zu Lenze Produkten finden Sie im Download-Bereich unter:

http://ww.Lenze.com

Zielgruppe

Diese Dokumentation richtet sich an Personen, die die Vernetzung von Geräten im Rahmen der Steuerungstechnik projektieren, installieren, in Betrieb nehmen und warten.

1.1 Dokumenthistorie

Material-Nr.	Version			Beschreibung
13294525	1.0	05/2009	TD17	Erstausgabe
13319345	2.0	10/2009	TD17	Allgemeine Überarbeitung
13369327	2.1	01/2011	TD17	Aktualisierung für Steuerungstechnik Release 2.5
13383677	2.2	07/2011	TD17	Kap. Fehlermeldungen bei nicht vorhandener Kommunikationskarte MC-PBM (📖 62) ergänzt.

Ihre Meinung ist uns wichtig!

Wir erstellten diese Anleitung nach bestem Wissen mit dem Ziel, Sie bestmöglich beim Umgang mit unserem Produkt zu unterstützen.

Vielleicht ist uns das nicht überall gelungen. Wenn Sie das feststellen sollten, senden Sie uns Ihre Anregungen und Ihre Kritik in einer kurzen E-Mail an:

feedback-docu@Lenze.de

Vielen Dank für Ihre Unterstützung.

Ihr Lenze-Dokumentationsteam

1.2 Verwendete Konventionen

Diese Dokumentation verwendet folgende Konventionen zur Unterscheidung verschiedener Arten von Information:

Informationsart	Auszeichnung	Beispiele/Hinweise
Zahlenschreibweise		
Dezimaltrennzeichen	Punkt	Es wird generell der Dezimalpunkt verwendet. Zum Beispiel: 1234.56
Textauszeichnung		
Versionsinfo	Textfarbe blau	Alle Informationen, die nur für oder ab einem bestimmten Softwarestand gelten, sind in dieser Dokumentation entsprechend gekennzeichnet. Beispiel: Diese Funktionserweiterung ist ab dem Softwarestand V3.0 verfügbar!
Programmname	» «	Die Lenze PC-Software »Engineer«...
Fensterbereich	*kursiv*	Das *Meldungsfenster*... / Das Dialogfeld *Optionen*...
Variablenbezeichner		Durch Setzen von *bEnable* auf TRUE...
Steuerelement	**fett**	Die Schaltfläche **OK**... / Der Befehl **Kopieren**... / Die Registerkarte **Eigenschaften**... / Das Eingabefeld **Name**...
Folge von Menübefehlen		Sind zum Ausführen einer Funktion mehrere Befehle nacheinander erforderlich, sind die einzelnen Befehle durch einen Pfeil voneinander getrennt: Wählen Sie den Befehl **Datei→Öffnen**, um...
Tastaturbefehl	**<fett>**	Mit **<F1>** rufen Sie die Online-Hilfe auf.
		Ist für einen Befehl eine Tastenkombination erforderlich, ist zwischen den Tastenbezeichnern ein "+" gesetzt: Mit **<Shift>+<ESC>**...

Programmcode	`Courier`	`IF var1 < var2 THEN`
		`  a = a + 1`
		`  END IF`

Schlüsselwort	**`Courier fett`**	
Hyperlink	unterstrichen	Optisch hervorgehobener Verweis auf ein anderes Thema. Wird in dieser Dokumentation per Mausklick aktiviert.
Symbole		
Seitenverweis	(📖 8)	Optisch hervorgehobener Verweis auf eine andere Seite. Wird in dieser Dokumentation per Mausklick aktiviert.
Schrittweise Anleitung		Schrittweise Anleitungen sind durch ein Piktogramm gekennzeichnet.

1.3 Verwendete Begriffe

Begriff	Bedeutung
»Engineer«	Engineering-Werkzeuge von Lenze, die Sie im gesamten Lebenszyklus einer Maschine - von der Planung bis zur Wartung - unterstützen.
»Global Drive Control« / »GDC«	
»PLC Designer«	
Codestelle	"Container" für einen oder mehrere Parameter, mit denen Sie Lenze Servo Drives parametrieren oder überwachen können.
Subcodestelle	Enthält eine Codestelle mehrere Parameter, so sind diese in sogenannten "Subcodestellen" abgelegt. In der Dokumentation wird als Trennzeichen zwischen der Angabe der Codestelle und der Subcodestelle der Schrägstrich "/" verwendet (z. B. "C00118/3").
IPC	Industrie-PC
PLC	Programmable Logic Controller (deutsche Bezeichnung: SPS)
GSD	Gerätestammdatendatei (Gerätebeschreibung für PROFIBUS-Geräte)

1.4 Verwendete Hinweise

Um auf Gefahren und wichtige Informationen hinzuweisen, werden in dieser Dokumentation folgende Signalwörter und Symbole verwendet:

Sicherheitshinweise

Aufbau der Sicherheitshinweise:

Piktogramm und Signalwort!

(kennzeichnen die Art und die Schwere der Gefahr)

Hinweistext

(beschreibt die Gefahr und gibt Hinweise, wie sie vermieden werden kann)

Piktogramm	Signalwort	Bedeutung
	Gefahr!	**Gefahr von Personenschäden durch gefährliche elektrische Spannung** Hinweis auf eine unmittelbar drohende Gefahr, die den Tod oder schwere Verletzungen zur Folge haben kann, wenn nicht die entsprechenden Maßnahmen getroffen werden.
	Gefahr!	**Gefahr von Personenschäden durch eine allgemeine Gefahrenquelle** Hinweis auf eine unmittelbar drohende Gefahr, die den Tod oder schwere Verletzungen zur Folge haben kann, wenn nicht die entsprechenden Maßnahmen getroffen werden.
	Stop!	**Gefahr von Sachschäden** Hinweis auf eine mögliche Gefahr, die Sachschäden zur Folge haben kann, wenn nicht die entsprechenden Maßnahmen getroffen werden.

Anwendungshinweise

Piktogramm	Signalwort	Bedeutung
	Hinweis!	Wichtiger Hinweis für die störungsfreie Funktion
	Tipp!	Nützlicher Tipp für die einfache Handhabung
		Verweis auf andere Dokumentation

2 Sicherheitshinweise

Beachten Sie die folgenden Sicherheitshinweise, wenn Sie mit dem Industrie-PC einen Antriebsregler bzw. eine Anlage in Betrieb nehmen möchten.

Lesen Sie die zu den System-Komponenten mitgelieferte Dokumentation sorgfältig durch, bevor Sie mit der Inbetriebnahme der Geräte und des Industrie-PCs beginnen!

Das Systemhandbuch enthält Sicherheitshinweise, die Sie beachten müssen!

Gefahr!

Nach heutiger wissenschaftlicher Erkenntnis ist es nicht möglich, die absolute Fehlerfreiheit einer Software sicherzustellen.

Sie müssen Anlagen mit eingebauten Antriebsreglern ggf. mit zusätzlichen Überwachungs- und Schutzeinrichtungen nach den jeweils gültigen Sicherheitsbestimmungen ausrüsten (z. B. Gesetz über technische Arbeitsmittel, Unfallverhütungsvorschriften), damit ein unzulässiger Betriebszustand zu keiner Gefährdung von Personen oder Einrichtungen führt.

Während der Inbetriebnahme dürfen sich keine Personen ohne ausreichenden Sicherheitsabstand in der Nähe des Motors oder den vom Motor angetriebenen Maschinenteilen aufhalten, da ansonsten eine Verletzungsgefahr durch bewegte Maschinenteile besteht.

Stop!

Wenn Sie in einem Engineering-Werkzeug Parameter verändern, während eine Online-Verbindung zum Gerät besteht, werden die Änderungen direkt in das Gerät übernommen!

Eine falsche Parametrierung kann zu nicht vorhersehbaren Motorbewegungen führen. Durch ungewollte Drehrichtung, zu hohe Geschwindigkeit oder ruckhaftem Lauf können angetriebene Maschinenteile beschädigt werden!

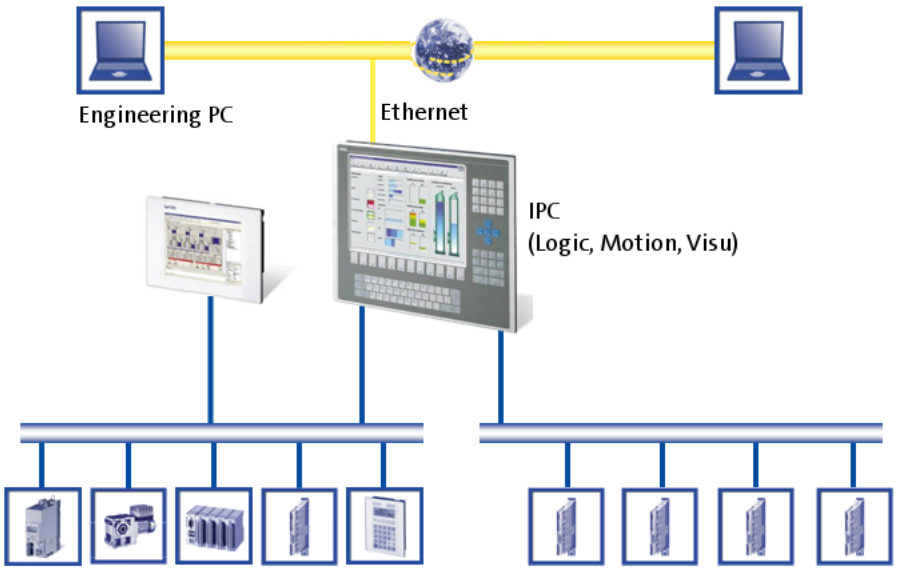
3 Das System "PC-based Automation"

Industrie-PCs (IPC) halten immer stärker Einzug in die Automatisierungstechnik. Industrie-PCs bieten wegen ihrer Skalierungsmöglichkeiten und den Kombinationsmöglichkeiten von Visualisierung und Steuerung auf einem Gerät für viele Anwendungen deutliche Vorteile.

Lenze-Industrie-PCs sind in den folgenden Software-Ausstattungen erhältlich:

- Industrie-PC als Komponente, auf Wunsch mit Betriebssystem, ohne weitere Software
- Industrie-PC als Visualisierungssystem
- Industrie-PC als Steuerungs- und Visualisierungssystem

Das System "PC-based Automation" ermöglicht die zentrale Steuerung von Logic- und Motion-Systemen.



Dafür stellt Lenze aufeinander abgestimmte Systemkomponenten zur Verfügung:

- Industrie-PCs als Steuerungs- und Visualisierungssystem
  - Der IPC ist die zentrale Komponente der PC-based Automation, der mit Hilfe der Runtime Software die Logic- und Motion-Funktionalitäten steuert.
  - Der IPC kommuniziert über den Feldbus mit den Feldgeräten.
  - Die IPCs sind in unterschiedlichen Bauformen lieferbar.

Hinweis!

Zum System "PC-based Automation" gehört zudem die HMI-Reihe EL 1xx PLC. Diese Geräte unterscheiden sich hinsichtlich der Leistungsfähigkeit und diversen anderen Details deutlich von den Industrie-PCs. Dennoch sind die Geräte der HMI-Reihe EL 1xx PLC in der Lage, kleinere Steuerungsfunktionen zu erfüllen.

- Engineering-Werkzeuge für den Engineering PC
  - Der Engineering-PC kommuniziert über Ethernet mit dem IPC.
  - Mit den verschiedenen Engineering-Werkzeugen (📖 22) konfigurieren und parametrieren Sie das System.
- Feldbusse
- Feldgeräte

4 Das Lenze Steuerungssystem mit PROFIBUS

Hinweis!

Im Lenze Steuerungssystem wird ausschließlich die PROFIBUS Master-Funktionalität (Logic-Bus) unterstützt.

In diesem Kapitel erfahren Sie grundlegende Informationen über ...

- das Bussystem PROFIBUS im Lenze Steuerungssystem;
- den Aufbau des Lenze Steuerungssystems mit dem PROFIBUS-Master;
- die notwendigen Komponenten zur PROFIBUS-Kommunikation.

4.1 Kurzbeschreibung PROFIBUS

PROFIBUS ist heute das am meisten verbreitete Feldbussystem. Mit der größten Auswahl an unterschiedlichen Feldgeräten wird der PROFIBUS moderneren Bussystemen gegenüber gelegentlich vorgezogen. Aufgrund der geringen Bandbreite und Synchronisierungsmechanismen wird der PROFIBUS im Rahmen der "PC-based Automation" nur als Logic-Bus angeboten.

Für folgende Anwendungen ist der Einsatz von PROFIBUS zu empfehlen:

- Steuerung von Anlagenteilen, die bereits zuvor mit PROFIBUS und einer anderen Steuerung automatisiert worden sind.
- Einsatz von Feldgeräten, die für andere Bussysteme (z. B. CANopen oder EtherCAT) nicht verfügbar sind.
- Kombination von PROFIBUS als Logic-Bus mit CANopen als Motion-Bus (📖 18)

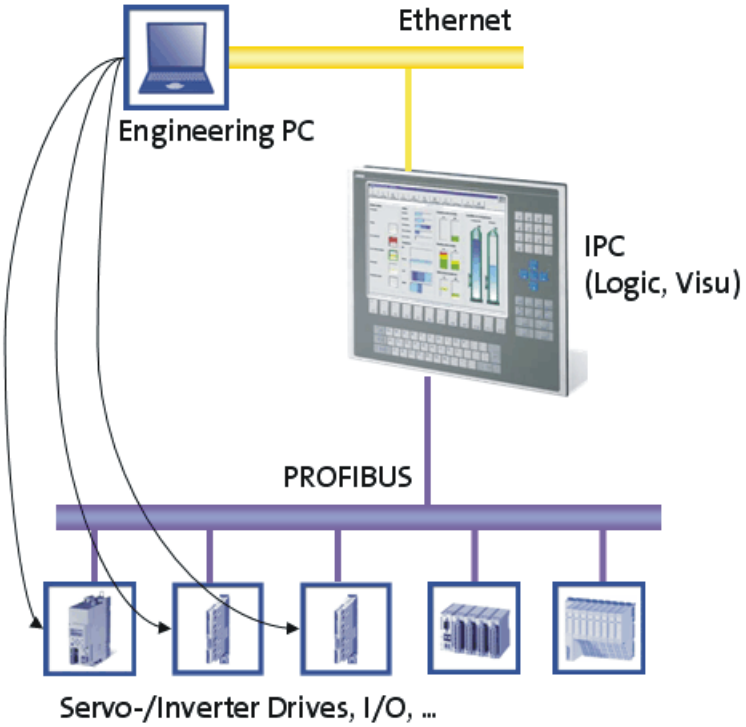
Tipp!

Ausführliche Informationen zum PROFIBUS finden Sie auf der Internet-Seite der PROFIBUS Nutzerorganisation:

www.profibus.com

4.1.1 Aufbau des PROFIBUS-Systems

Prinzipieller Aufbau



Physikalischer Aufbau

- Der Industrie-PC (IPC) ist der PROFIBUS-Master. Er kann mit einem oder mehreren Teilnehmern (Slaves) kommunizieren.
- Intern besitzt der PROFIBUS eine Linientopologie (ohne Repeater) oder eine Baumtopologie (mit Repeater).
  ▸ Prinzipielle Verdrahtung des PROFIBUS (📖 17)
- Das PROFIBUS-Netzwerk muss am ersten und letzten Teilnehmer abgeschlossen sein. Der Busabschluss-Widerstand ist im Busanschluss-Stecker eingebaut und wird mit einem Schalter aktiviert.

Parametrierung

Die Parametrierung der PROFIBUS-Teilnehmer ist auf unterschiedliche Weise möglich:

- Direkter Zugang der Engineering-Software (vom Engineering PC aus) zum Slave-Feldgerät. Je nach Gerätetyp über folgende Schnittstellen:
  - CAN
  - Ethernet
  - LECOM
  - Diagnoseschnittstelle
- Parametertransfer von der Steuerung (der Parametertransfer ist manuell zu programmieren)
  - Die PROFIBUS-Konfiguration erfolgt ausschließlich mit dem »PLC Designer«: Aufbau der Steuerungskonfiguration und Erstellung des PLC-Programms
  - Über Gerätestammdatendateien (GSD-Dateien) können alle PROFIBUS-Slaves angeschaltet werden.

Hinweis!

Die Funktion "IPC als Gateway" ist in Verbindung mit PROFIBUS nicht verfügbar.

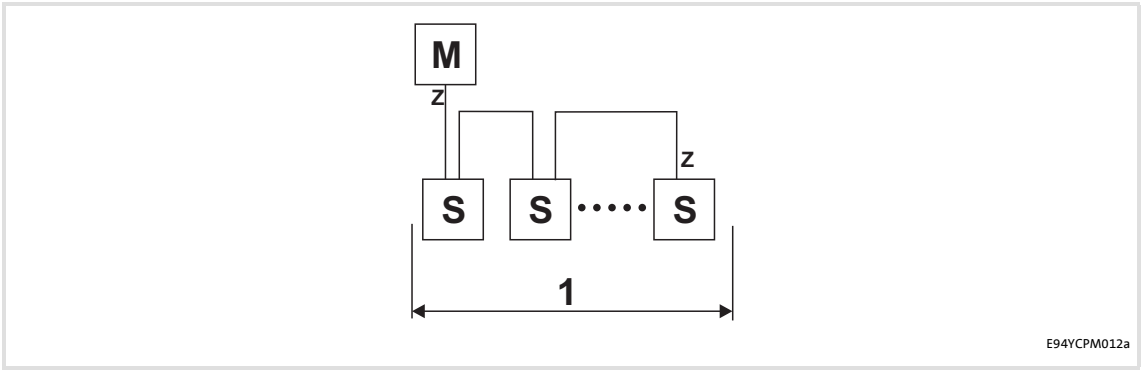
"Online gehen" mit dem »Engineer« oder dem »GDC« über den IPC als Gateway ist daher nicht möglich.

4.1.2 Prinzipielle Verdrahtung des PROFIBUS

In den folgenden Beispielen sind zwei einfache PROFIBUS-Netzwerke dargestellt.

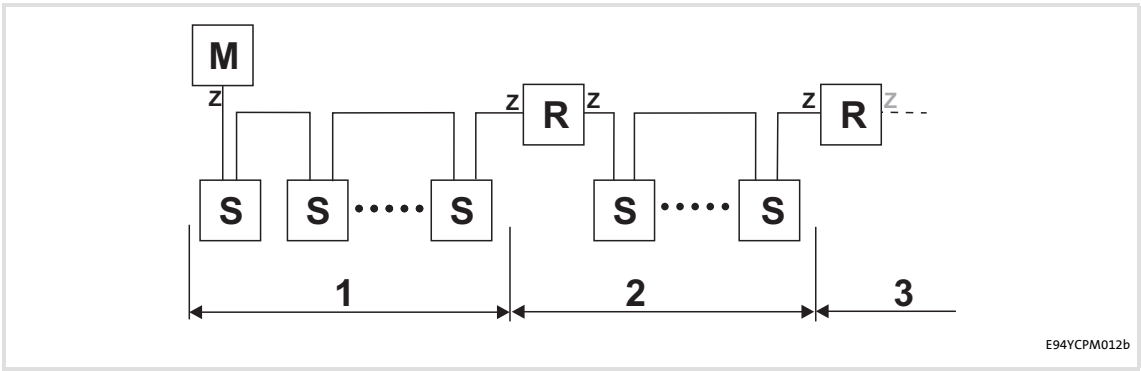
Jedes Segment des Netzwerkes muss an seinem Anfang und an seinem Ende abgeschlossen sein. Die Bus-Abschlüsse des PROFIBUS sind in jedem der folgenden Beispiele mit einem "Z" gekennzeichnet.

Bei einem PROFIBUS-Netzwerk aus nur einem Segment bildet der PROFIBUS-Master (M) mit integriertem Bus-Abschluss den Anfang, während am Ende der Stecker des letzten Gerätes (S) den Busabschluss hat.



[4-1] PROFIBUS-Netzwerk mit einem Segment

Ein aus mehreren Segmenten bestehendes PROFIBUS-Netzwerk enthält Repeater (R) zur Kopplung der Segmente. Die Repeater haben integrierte Busabschlüsse.



[4-2] PROFIBUS-Netzwerk mit Repeater

Sollte am Ende des Segments kein Repeater eingesetzt werden, muss der Busabschluss im Stecker des letzten Gerätes aktiviert werden.

Hinweis!

Repeater besitzen keine Geräteadresse. Bei der Berechnung der maximalen Teilnehmeranzahl reduzieren sie aber auf jeder Segmentseite die Teilnehmeranzahl um 1.

Mit Repeatern können Linien- und Baumtopologien aufgebaut werden. Die maximale Gesamtausdehnung des Bussystems ist abhängig von der verwendeten Übertragungsrate und der Repeater-Anzahl.

4.1.3 Kombination mit anderen Bussystemen

Das Bussystem PROFIBUS kann mit CANopen kombiniert werden. Dies ist sinnvoll, wenn nicht alle Feldgeräte für das gleiche Bussystem verfügbar sind oder parallel zum PROFIBUS (als Logic-Bus) ein Motion-Bus (CANopen) benötigt wird. Die Bussysteme werden in der Steuerung synchronisiert.

Hinweis!

- Ein Mischbetrieb ist nur bei Industrie-PCs möglich, die zwei Erweiterungs-Schächte für Kommunikationskarten besitzen. Bei der "Command Station" ist der Mischbetrieb nicht möglich.
- Im Release 2.5 kann PROFIBUS nicht mit EtherCAT kombiniert werden.
- In der Steuerungskonfiguration muss der PROFIBUS-Master an erster Position – vor den CANopen Motion-Teilnehmern – angeordnet werden.

4.1.4 Feldgeräte

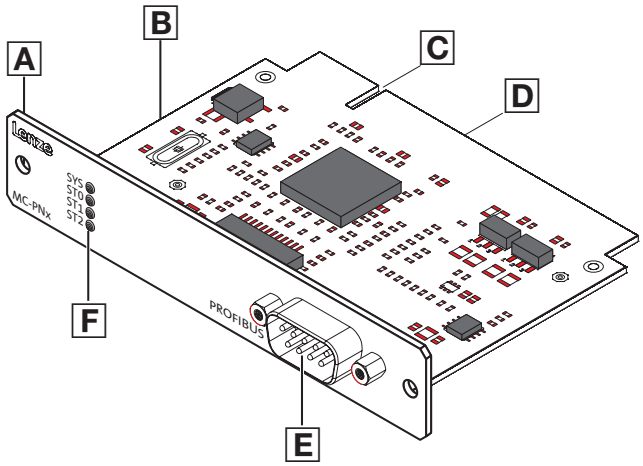
Das Lenze Steuerungssystem unterstützt für den PROFIBUS folgende Logic-Komponenten:

Grundgeräte		PROFIBUS Kommunikationskarten/-module
Industrie-PCs	EL x1xx PLC	MC-PBM (PROFIBUS Master)
	EL x8xx	
	CS x8xx	
	CPC x8xx	
Servo Drives 9400	HighLine	E94AYCPM
	PLC	
Inverter Drives 8400	BaseLine	E84AYCPM
	StateLine	
	HighLine	
	TopLine	
I/O-System IP20	EPM-T120	
	EPM-T121	
I/O-System 1000	EPM-S120 (*in Vorbereitung*)	
Frequenzumrichter	8200 vector	E82ZAFPCxxx
Servo System ECS	ECSxE	EMF2133IB
	ECSXS (Speed & Torque)	
	ECSxP (Posi & Shaft)	
	ECSxA (Application)	

4.2 PROFIBUS-Hardware für den Industrie-PC

Kommunikationskarte MC-PBM

Die Kommunikationskarte MC-PBM ist eine Einsteckkarte zur Anbindung eines Industrie-PCs als PROFIBUS-Master an ein PROFIBUS-Netzwerk.



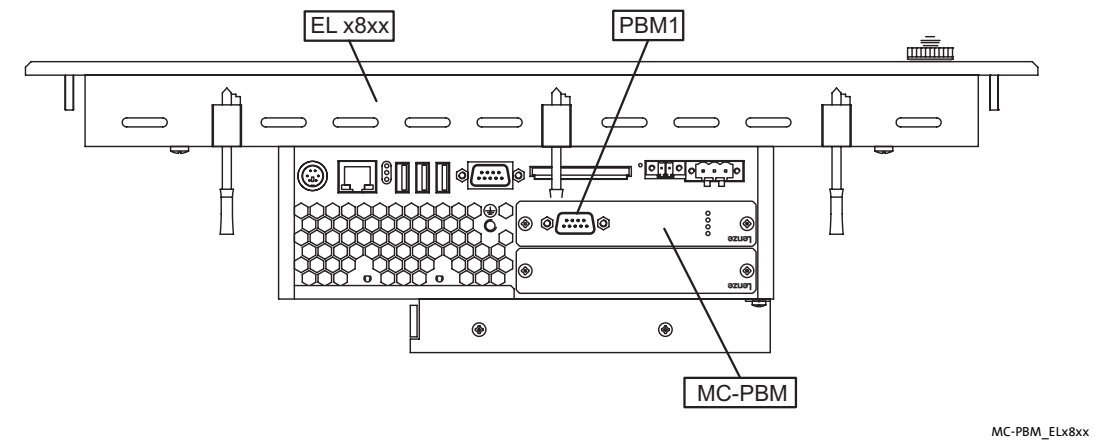
A Frontblende
B Platine
C Codierung
D Anschluss Industrie-PC
E Anschluss PROFIBUS (SUB-D, 9-pol. Stecker) (📖 20)
F LED-Statusanzeigen (📖 63)

▸ Technische Daten der Kommunikationskarte MC-PBM (📖 20)

Verwendungsmöglichkeiten

Die Kommunikationskarte MC-PBM kann in Steckplatz 1 und in Steckplatz 2 des Industrie-PCs eingebaut sein. Pro Industrie-PC sind mehrere PROFIBUS-Kommunikationskarten verwendbar.

Beispiel: Industrie-PC EL x8xx mit MC-PBM in Steckplatz 1



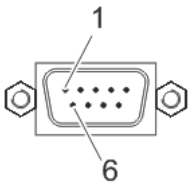
Legende	
EL x8xx	Industrie-PC der Reihe EL x8xx
PBM1	PROFIBUS-Anschluss
MC-PBM	Kommunikationskarte PROFIBUS-Master

5 Technische Daten

5.1 Technische Daten der Kommunikationskarte MC-PBM

Bereich	Werte
Protokoll	PROFIBUS-DP (V0, V1), ISO 7498
Kommunikationsmedium	RS485
Netzwerktopologie	Beidseitig abgeschlossene Linie (ohne Repeater) / Baum (mit Repeater) • Abschluss mit Sub-D-Stecker
Typ innerhalb des Netzwerks	Master
Max. Anzahl Teilnehmer pro Segment	63
Max. Anzahl Teilnehmer pro Netz	128
Übertragungsrate	Siehe Kap. "Übertragungsrate / Länge des Buskabels" (📖 21)
Buslänge	
Anschluss	SUB-D, 9-pol. Stecker

Anschluss PROFIBUS (SUB-D, 9-pol. Stecker)

Ansicht	Pin	Belegung	Beschreibung
	1	frei	-
	2	frei	-
	3	RxD/TxD-P	Datenleitung-B (Empfangs-/Sendedaten-Plus)
	4	RTS	Request To Send (Empfangs-/Sendedaten, kein Differenzsignal)
	5	M5V2	Datenbezugspotenzial (Masse zu 5 V)
	6	P5V2	5 V DC / 30 mA (Busabschluss)
	7	frei	-
	8	RxD/TxD-N	Datenleitung-A (Empfangsdaten-/Sendedaten-Minus)
	9	frei	-

5.2 Spezifikation des Buskabels

Folgen Sie bei der Verwendung des Buskabels den Angaben der PROFIBUS-Nutzerorganisation.

Bereich	Werte
Leitungswiderstand	135 ... 165 Ω/km, (f = 3 ... 20 MHz)
Kapazitätsbelag	≤ 30 nF/km
Schleifenwiderstand	< 110 Ω/km
Aderdurchmesser	> 0.64 mm
Aderquerschnitt	> 0.34 mm^2
Adern	2-fach verdrillt, isoliert und abgeschirmt

Übertragungsrate / Länge des Buskabels

Passen Sie die Übertragungsrate ensprechend der Länge des Buskabels an.

Übertragungsrate [kBit/s]	Max. Buslänge [m]
9.6 ... 93.75	1200
187.5	1000
500	200
1500	200
3000 ... 12000	100

Hinweis!

Die von Datenmenge, Zykluszeit und Teilnehmeranzahl abhängige Übertragungsrate sollte nur so hoch gewählt werden, wie es für die Anwendung erforderlich ist.

6 Inbetriebnahme des PROFIBUS

In diesem Kapitel erfahren Sie, wie Sie das Lenze Steuerungssystem mit PROFIBUS in Betrieb nehmen.

Je nach verwendeten Feldgeräten sind folgende Lenze Engineering-Werkzeuge erforderlich:

- »PLC Designer«
- »Engineer«
- »Global Drive Control« (GDC)

Tipp!

Zur Verwendung weiterer Feldbussysteme kann weitere Engineering-Software erforderlich sein. Weiterführende Informationen finden Sie in den entsprechenden Kommunikationshandbüchern.

6.1 Übersicht der Inbetriebnahmeschritte

Die Haupt-Inbetriebnahmeschritte sind in der folgenden Tabelle zusammengefasst:

Schritt	Tätigkeit	Zu verwendende Software
1.	Bustopologie planen (📖 23)	
2.	Feldgeräte installieren (📖 23)	
3.	Projektordner anlegen (📖 24)	
4.	Gerätestammdatendateien (GSD) ablegen (📖 24)	
5.	Feldgeräte in Betrieb nehmen (📖 25)	»Engineer« »GDC«
6.	PLC-Programm anlegen (📖 26)	»PLC Designer«
7.	PROFIBUS-Master konfigurieren (📖 29)	»PLC Designer«
8.	PROFIBUS-Slave konfigurieren (📖 33)	»PLC Designer«
9.	Projektdaten übersetzen (📖 36)	»PLC Designer«
10.	Mit »PLC Designer« in die Steuerung einloggen (📖 36)	»PLC Designer«
11.	PLC-Programm laden und starten (📖 36)	»PLC Designer«

6.2 Detaillierte Inbetriebnahmeschritte

In den folgenden Abschnitten werden die einzelnen Inbetriebnahmeschritte beschrieben.

Folgen Sie den Anweisungen Schritt-für-Schritt, um Ihr System in Betrieb zu nehmen.

Weiterführende Informationen zum Umgang mit den Lenze Engineering-Werkzeugen finden Sie in den entsprechenden Handbüchern und Online-Hilfen.

6.2.1 Bustopologie planen

Bevor Sie ein PROFIBUS-Netzwerk aufbauen, erstellen Sie einen Plan Ihres Netzwerks.

Hinweis!

Beachten Sie die Abhängigkeit zwischen Busleitungslänge und Übertragungsrate.

▸ Übertragungsrate / Länge des Buskabels (📖 21)

So planen Sie die Bustopologie Ihrer Konfiguration:

1. Legen Sie ein Übersichtsbild des geplanten PROFIBUS-Netzwerks mit allen einzubindenden Feldgeräten an.
2. Beginnen Sie dabei mit dem Industrie-PC (Master).
3. Ordnen Sie darunter die weiteren Feldgeräte (Slaves) an.

6.2.2 Feldgeräte installieren

Installieren Sie die Feldgeräte gemäß den Angaben in den gerätespezifischen Montageanleitungen.

Bus-Abschlusswiderstand aktivieren

Das PROFIBUS-Netzwerk muss am ersten und letzten Teilnehmer abgeschlossen sein. Der Busabschluss-Widerstand ist im Busanschluss-Stecker eingebaut und wird mit einem Schalter aktiviert.

Hinweis!

Falls einzelne Busteilnehmer abgeschaltet werden, muss dafür gesorgt werden, dass die Busabschlüsse an den physikalischen Leitungsenden weiter aktiv bleiben.

Der Busabschluss ist nicht mehr aktiv, wenn z. B. im Servicefall der Stecker abgezogen wurde oder die Baugruppenversorgung abgeschaltet wurde.

6.2.3 Projektordner anlegen

Legen Sie einen Projektordner auf dem Engineering-PC an.

Speichern Sie in diesem Projektordner die in den nachfolgenden Projektierungsschritten erzeugten Daten:

- Im »Engineer« oder »GDC« erstellte Projektdaten
- Im »PLC Designer« erstellte Projektdatei
- Projektdaten anderer Engineering-Werkzeuge

Tipp!

Erstellen Sie für jede PROFIBUS-Konfiguration einen separaten Projektordner zur Aufnahme der Projektdateien.

6.2.4 Gerätestammdatendateien (GSD) ablegen

In einer GSD-Datei sind die für die übergeordnete Steuerung erforderlichen Daten der Busperipherie abgelegt. Diese Datei wird bei der Programmierung der Steuerung benötigt.

Aktuelle GSD-Dateien (z. B. zum I/O-System IP20 oder zu Lenze PROFIBUS-Kommunikationsbaugruppen) können Sie im Download-Bereich unter http://www.Lenze.com herunterladen.

Speichern Sie die notwendigen GSD-Dateien im »PLC Designer« Targets-Verzeichnis:

- C:\...\Lenze\PlcDesigner\Targets\L-force_Logic_x800_V8\PlcConfPCMatic
- C:\...\Lenze\PlcDesigner\Targets\L-force_Motion_x800_V8\PlcConfPCMaticMotion

Hinweis!

Der PROFIBUS kann in Motion- und Logic-Systemen eingesetzt werden. Im Lenze Steuerungssystem ist der PROFIBUS ausschließlich der Logic-Bus. Verwenden Sie den PROFIBUS in einem Motion-System nur, wenn ein zusätzlicher Motion-Bus (z. B. CAN) eingesetzt wird.

▸ CANopen mit PROFIBUS (📖 37)

Tipp!

Die Ablage der GSD-Dateien ist auch über den »PLC Designer« mit dem Menübefehl **Extras→Konfigurationsdatei hinzufügen** möglich.

6.2.5 Feldgeräte in Betrieb nehmen

Parametrieren Sie die am PROFIBUS angeschlossenen Lenze-Feldgeräte – je nach Gerät – entweder mit dem »Engineer« oder dem »GDC«.

Die PROFIBUS-Konfiguration erfolgt ausschließlich mit dem »PLC Designer«.

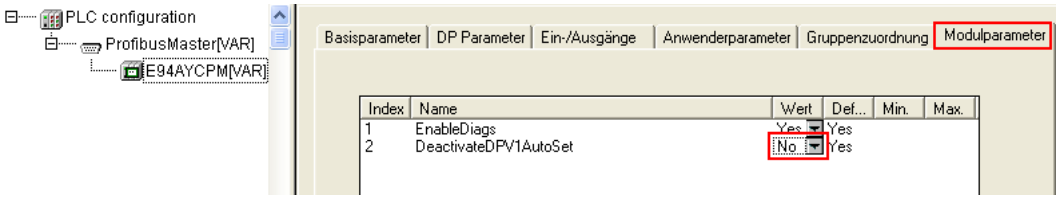
Beachten Sie die Informationen zur Inbetriebnahme in den Dokumentationen der Feldgeräte.

Tipp!

Wir empfehlen, jedes Feldgerät einzeln in Betrieb zu nehmen und dann in das PLC-Programm einzubinden.

Servo Drives 9400 HighLine

Bei der Inbetriebnahme eines Servo Drive 9400 HighLine muss der Modulparameter **DeactivateDPV1AutoSet** auf den Wert **"No"** eingestellt werden:



6.2.6 Konfiguration im »PLC Designer«

Mit dem »PLC Designer« bilden Sie die Feldgeräte-Topologie in der Steuerungskonfiguration ab.

Tipp!

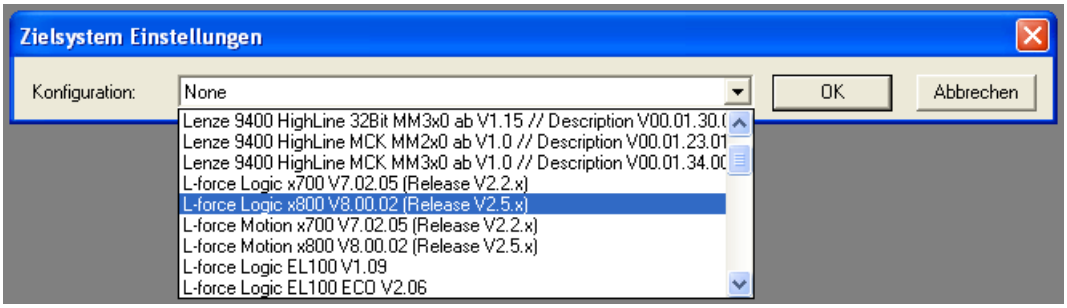
Im »PLC Designer« sind PROFIBUS-Teilnehmer sowie Teilnehmer an anderen Feldbus-Systemen konfigurierbar.

▸ CANopen mit PROFIBUS (📖 37)

6.2.6.1 PLC-Programm anlegen

So legen Sie ein PLC-Programm im »PLC Designer« an:

1. Neues »PLC Designer«-Projekt anlegen:
   - Menübefehl: **Datei→Neu**
2. Im Dialogfenster *Zielsystem Einstellungen* das passende Zielsystem auswählen:



Zielsystem	Verwendung mit Industrie-PC Reihe			
	EL x8xx	CS x8xx	CPC x8xx	EL 1xx PLC
L-force Logic x800 V8.xx.xx	●	●	●	-
L-force Logic EL1xx V1.xx	-	-	-	●

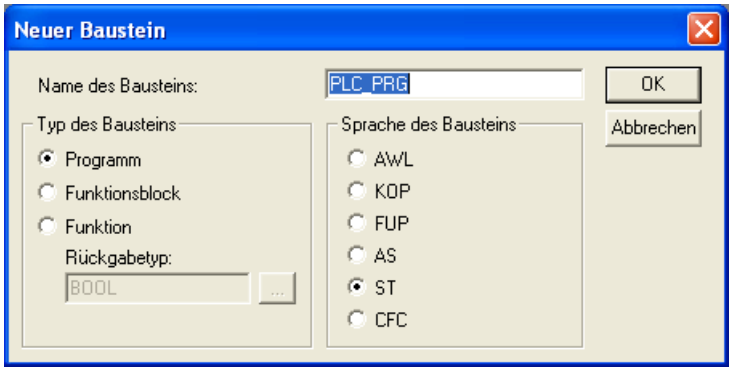
Die Zielsysteme des Release 2.2/2.3 (L-force Logic x700) sind auch für die Gerätereihen **EL x8xx**, **CS x8xx** und **CPC x8xx** verwendbar.

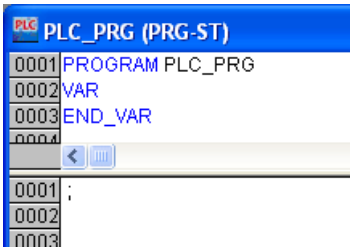
3. Konfiguration der Zielsystem-Einstellung mit der Schaltfläche **OK** bestätigen.

4. Baustein anlegen:

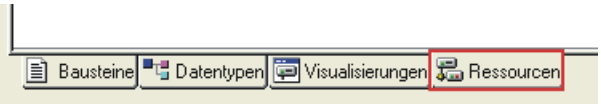
Hinweis!

Zur ordnungsgemäßen Funktion muss der Baustein mindestens eine Anweisung enthalten.

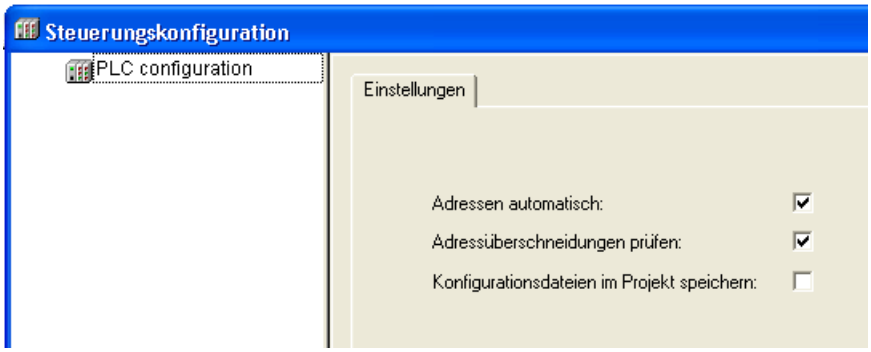




5. Steuerungskonfiguration anlegen:
   - Dialogfeld *Ressourcen* öffnen:



   - Dialogfenster *Steuerungskonfiguration* öffnen:



Einstellung	Beschreibung
Adressen automatisch	Jedes neu hinzugefügte Modul erhält automatisch eine Adresse, die sich aus der des zuvor eingefügten Moduls plus dessen Größe ergibt. Wird ein Modul aus der Konfiguration entfernt, werden die Adressen der nachfolgenden Module automatisch angepasst. Über den Menübefehl **Extras→Adressen berechnen** werden die Adressen ab dem aktuell ausgewählten Knoten (Modul) neu ermittelt.
Adressüberschneidungen prüfen	Adressüberschneidungen werden beim Übersetzen des Projektes überprüft und gemeldet.
Konfigurationsdateien im Projekt speichern	Die Information der Konfigurationsdatei(en) *.cfg und der Gerätedateien, die der aktuellen Steuerungskonfiguration zugrunde liegen, wird im Projekt gespeichert.

Hinweis!

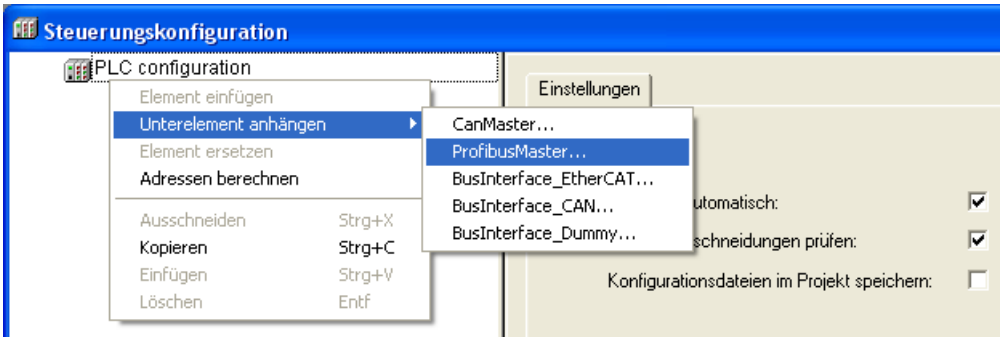
Wir empfehlen die Standardeinstellung beizubehalten. Bei manueller Adressvergabe müssen Sie darauf achten, dass in der gesamten Steuerungskonfiguration jede Objektadresse **eindeutig** ist.

Ausführliche Informationen dazu finden Sie in der Dokumentation/Online-Hilfe des »PLC Designer«.

6.2.6.2 PROFIBUS-Master konfigurieren

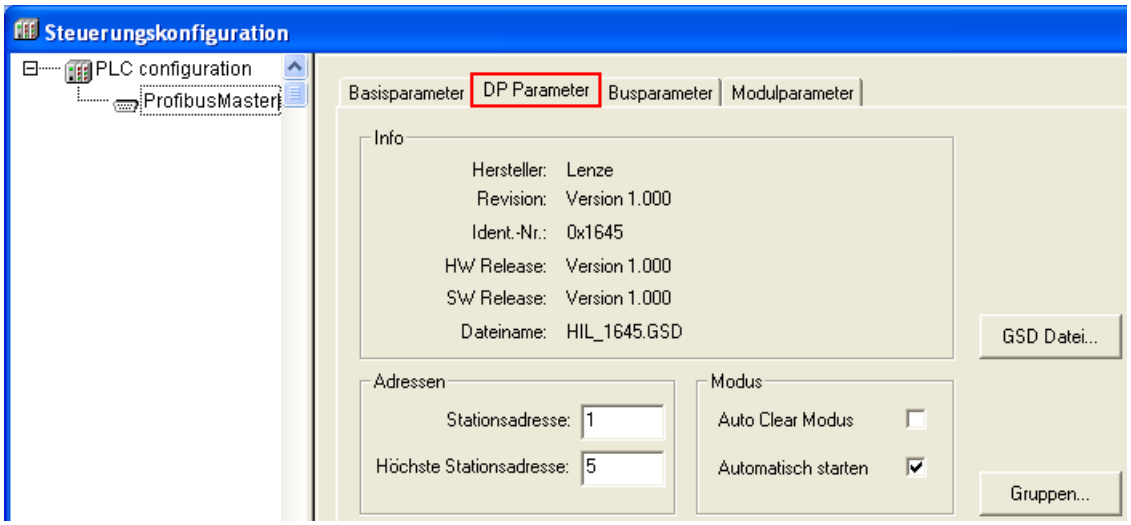
So konfigurieren Sie den PROFIBUS-Master:

1. Bus-Interface in die PLC-Konfiguration einfügen:

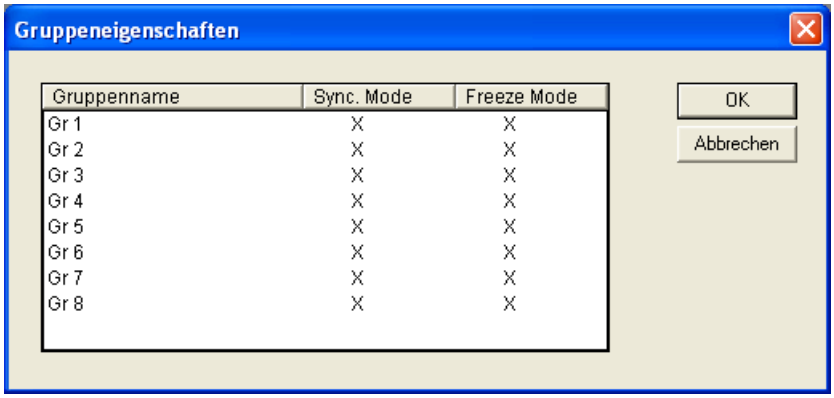


   Das Unterelement "ProfibusMaster" repräsentiert die PROFIBUS-Schnittstelle des IPC, an die der Logic-Bus angeschlossen ist.

2. DP Parameter für den PROFIBUS-Master einstellen:



   - Die Standardeinstellung der PROFIBUS-Master Stationsadresse ist ’1’. Ändern Sie die Einstellung nur, wenn die Adresse von ’1’ abweichen soll.
   - Die "Höchste Stationsadresse" wird mit jedem hinzugefügten Slave automatisch inkrementiert. Der Wert sollte daher nicht manuell geändert werden.
   - Über die Schaltfläche **GSD Datei...** kann die gerätezugehörige GSD-Datei geöffnet und eingesehen werden.
   - Die Schaltfläche **Gruppen...** führt zum Dialog *Gruppeneigenschaften*. Die Gruppeneigenschaften beziehen sich auf die dem Master zugeordneten Slaves.



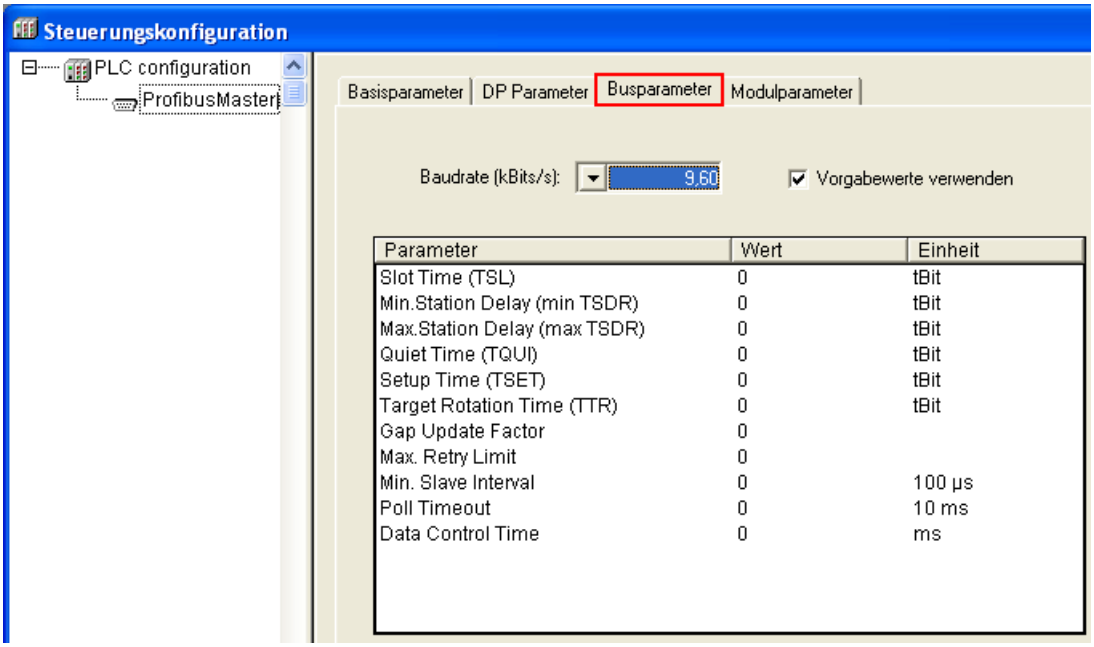
Bis zu acht Gruppen können eingerichtet werden. Stellen Sie für jede Gruppe ein, ob sie im Freeze-Mode und/oder Sync-Mode betrieben werden soll. Durch die Zuordnung der Slaves (siehe "Eigenschaften des DP-Slaves", "Gruppenzuordnung") zu verschiedenen Gruppen kann der Datenaustausch vom Master über ein Global-Control-Kommando synchronisiert werden.

Mit einem Freeze-Kommando veranlasst ein Master einen Slave oder eine Gruppe, die Eingänge im momentanen Zustand "einzufrieren" und diese Daten beim darauf folgenden Datenaustausch zu übertragen.

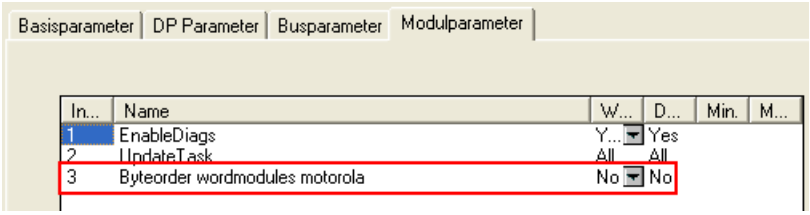
Mit einem Sync-Kommando werden die Slaves veranlasst, die im folgenden Datenaustausch vom Master empfangenen Daten mit dem nächsten Sync-Kommando zeitlich synchron an die Ausgänge durchzuschalten.

Zum Ein-/Ausschalten der Freeze- und Sync-Option für eine Gruppe, klicken Sie mit der linken Maustaste an entsprechender Stelle in der Tabelle, um bei der gewünschten Option ein 'X' zu platzieren/entfernen oder die rechte Maustaste, um die Option über ein Kontextmenü zu aktivieren/deaktivieren. Außerdem können Sie hier die Gruppennamen editieren.

3. Unter der Reiterkarte **Busparameter** die Baudrate für den PROFIBUS einstellen:



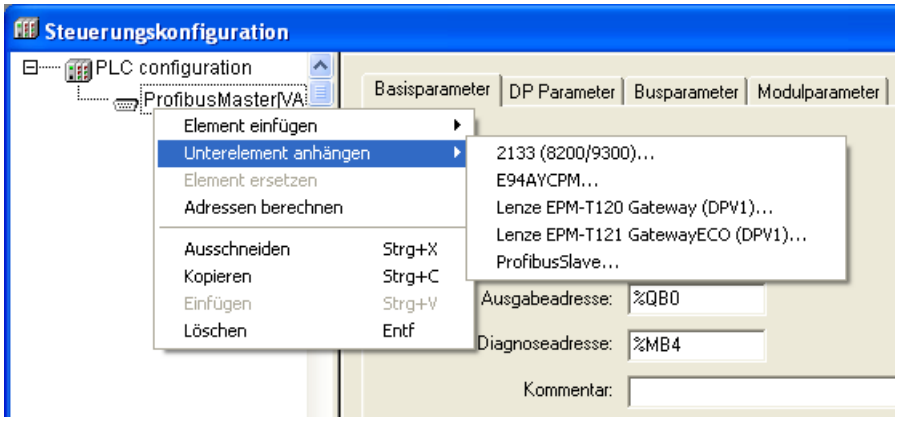
4. Unter der Reiterkarte **Modulparameter** den Parameter "Byteorder wordmodules motorola" einstellen:



Der Parameter "Byteorder wordmodules motorola" bestimmt wie die Daten vom PROFIBUS ins Prozessabbild kopiert werden.

Einstellung	Beschreibung
No	Standardeinstellung: Die Daten werden vom Bus im Motorola-Format (Big Endian) genommen und in das Intel-Format (Little Endian) kopiert. Dem Anwender stellt sich ein Integerwert (INT) korrekt dar (Beispiel: '0x1234' → '0x1234'). **Wir empfehlen, diese Einstellung beizubehalten. So werden die Slave-Daten korrekt in das Prozessabbild übernommen.**
Yes	Die Daten werden ohne Drehung vom Bus ins Prozessabbild kopiert. Dem Anwender stellt sich ein Integerwert (INT) gedreht dar (Beispiel: '0x1234' → '0x3412'). • Kompatibilität zu Projekten aus der Version 1.5

5. PROFIBUS-Slave anhängen:



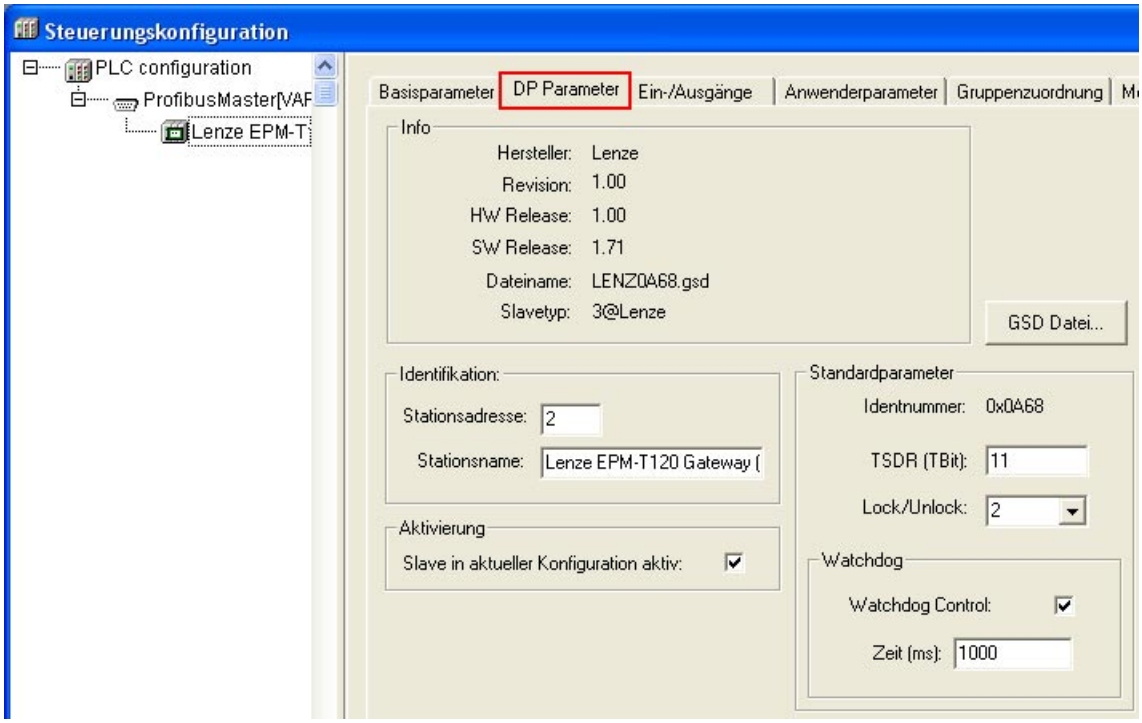
Die GSD-Datei des PROFIBUS-Slaves muss im »PLC Designer« Targets-Verzeichnis abgelegt sein.

▸ Gerätestammdatendateien (GSD) ablegen (📖 24)

6.2.6.3 PROFIBUS-Slave konfigurieren

So konfigurieren Sie den PROFIBUS-Slave:

1. DP Parameter für den PROFIBUS-Slave einstellen:

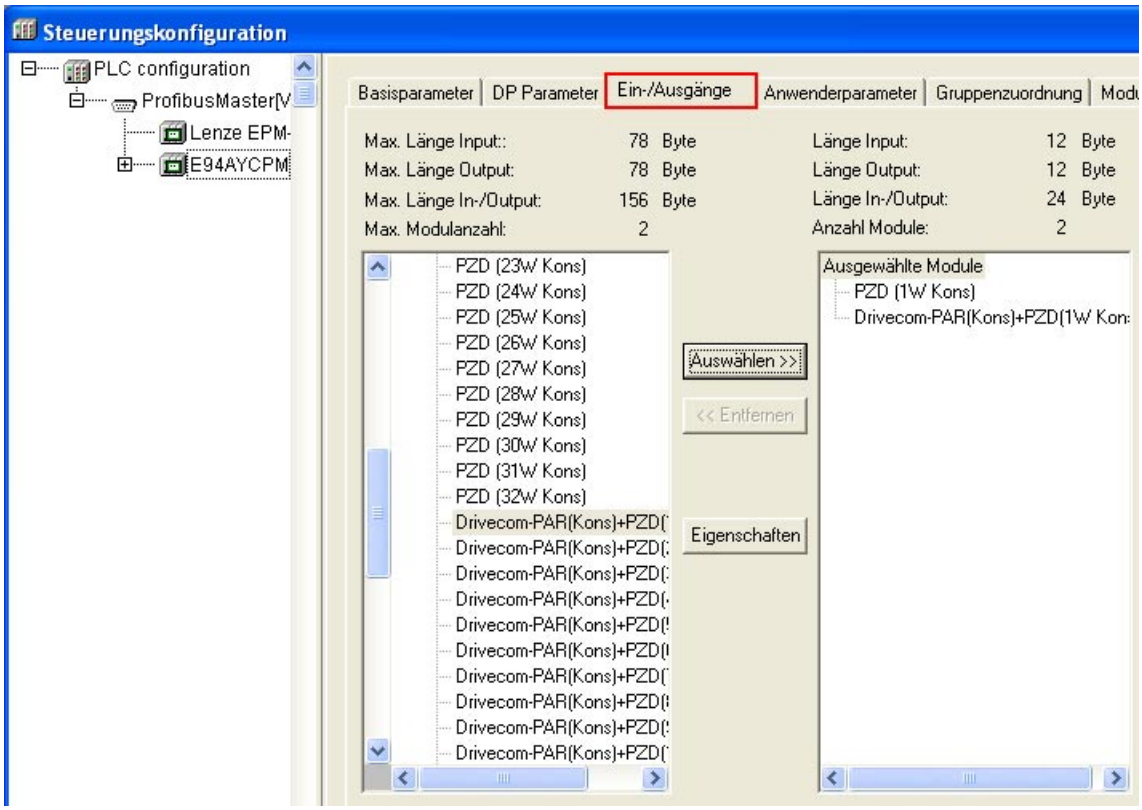


- Geben Sie hier die Stationsadresse des PROFIBUS-Slaves ein.
- Es muss keine Baudrate eingestellt werden, da der Slave die Baudrate automatisch erkennt.
- Über die Schaltfläche **GSD Datei...** kann die gerätezugehörige GSD-Datei geöffnet und eingesehen werden.

2. Den Slaves unter der Reiterkarte **Ein-/Ausgänge** die Ein- und Ausgabeobjekte konfigurieren:

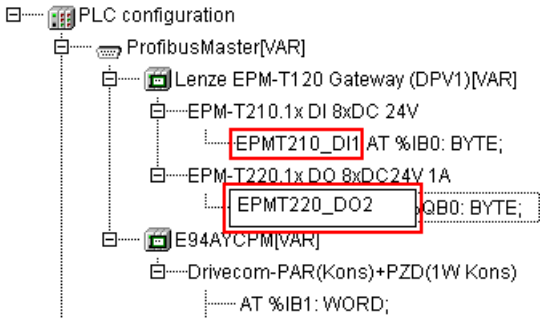
Hinweis!

Die Ein- und Ausgabeobjekte müssen in der Reihenfolge konfiguriert werden, wie sie physikalisch am Bus angeordnet sind.



- Der Dialog listet im linken Fenster alle in der GSD-Datei des Slaves verfügbaren Ein- und Ausgangsmodule, Prozessdatenobjekte (PZD) und DRIVECOM-Parameterobjekte (z. B. bei Servo Drives 9400) auf.
- Das Fenster rechts enthält die aktuell für dieses Gerät gewählte Konfiguration bzgl. der Ein- und Ausgänge.
- Nicht möglich ist diese Art der Auswahl bei nicht-modularen Slaves. Diese erzwingen unmittelbar eine geschlossene Darstellung ihrer Ein- und Ausgänge im rechten Fenster.
- Die Schaltfläche **Eigenschaften** führt zum Dialog *Moduleigenschaften* des aktuell in der linken oder rechten Liste angewählten Ein- oder Ausgabeobjekts.

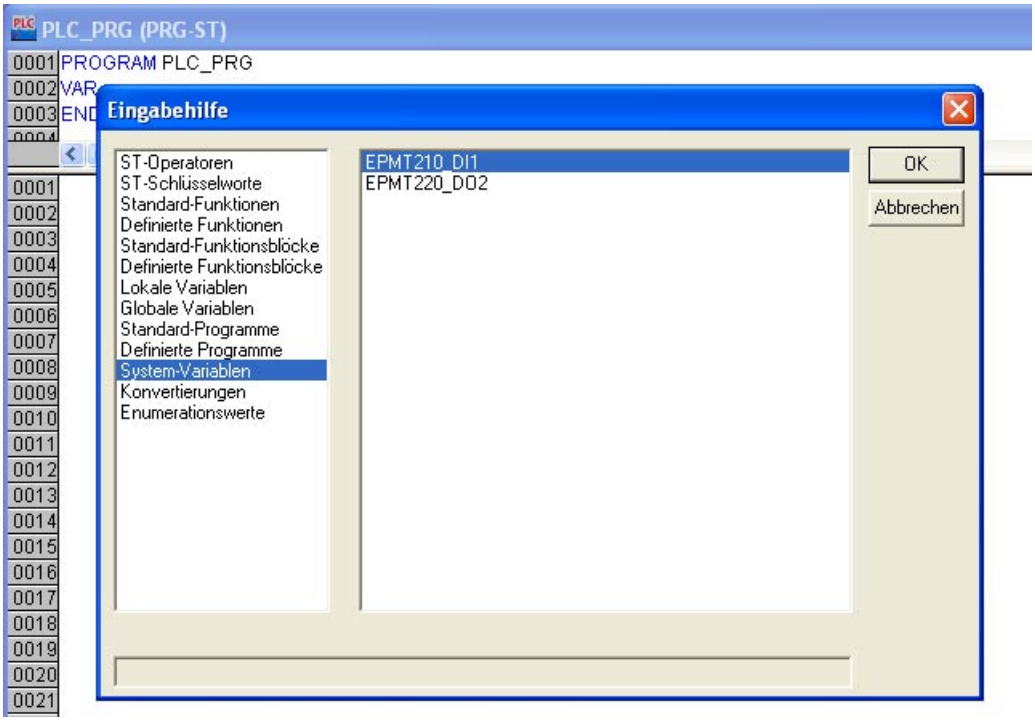
3. Vergeben Sie für jede Adresse der Ein- und Ausgabeobjekte (z. B. *%IB0*, *%QB0*, ...) einen in der gesamten Steuerungskonfiguration **eindeutigen** symbolischen Namen gemäß der IEC 61131-Syntax (keine Leerzeichen und führende Ziffern im Variablennamen):
   - Die Eingabe symbolischer Namen ist durch einen Mausklick vor 'AT %... ;' möglich.



Hinweis!

Durch die Eingabe symbolischer Namen werden entsprechende System-Variablen für das PLC-Programm erzeugt.

Verwenden Sie innerhalb des PLC-Programms immer die System-Variablen, um auf die Ein- und Ausgabeobjekte zuzugreifen oder ihnen Werte zuzuweisen.



Mit der Funktionstaste **<F2>** öffnen Sie die Eingabehilfe im »PLC Designer«.

6.2.6.4 Projektdaten übersetzen

Um die Projektdaten zu übersetzen, wählen Sie den Menübefehl **Projekt→Übersetzen**, oder betätigen sie die Funktionstaste **<F11>.**

- Traten beim Übersetzen Fehler auf, können Sie diese anhand der »PLC Designer«-Fehlermeldungen lokalisieren und entsprechend korrigieren. Übersetzen Sie danach die Projektdaten erneut.
- Wenn beim Übersetzten keine Fehler auftraten, speichern Sie das »PLC Designer«-Projekt im Projektordner.

6.2.6.5 Mit »PLC Designer« in die Steuerung einloggen

Um den »PLC Designer« in die Steuerung einzuloggen, wählen Sie den Menübefehl **Online→Einloggen**.

- Dazu muss dass PLC-Programm fehlerfrei sein.
- Den erscheinenden Abfragedialog, ob das neue Programm geladen werden soll, mit der Schaltfläche **Ja** bestätigen.

6.2.6.6 PLC-Programm laden und starten

So laden und starten Sie das PLC-Programm auf dem IPC:

1. Menübefehl **Online→Laden** auswählen.
2. Im erscheinenden Dialogfenster die gewünschte Datei auswählen.
3. Die Auswahl mit der Schaltfläche **Öffnen** bestätigen.
   - Die Datei wird in den IPC geladen und dort unter demselben Namen abgelegt.
   - Der PROFIBUS wird initialisiert.
4. Menübefehl **Online→Start** auswählen.
   - Das PLC-Programm wird ausgeführt.
   - Der zyklische Datentransfer beginnt.

Hinweis!

- Der Bus läuft an, auch wenn nicht alle Teilnehmer am Bus verfügbar sind.
- Wird das PLC-Programm gestoppt (Menübefehl **Online→Stop**) läuft der zyklische Datentransfer solange weiter, bis ein Reset-Ursprung ausgeführt wird (Menübefehl **Online→Reset (Ursprung)**).

Tipp!

Mit dem Menübefehl **Online→Datei aus Steuerung laden** können Sie eine auf dem IPC abgelegte Datei wieder in das »PLC Designer«-Projekt laden.

7 CANopen mit PROFIBUS

Das Bussystem PROFIBUS kann mit CANopen kombiniert werden. Dies ist sinnvoll, wenn nicht alle Feldgeräte für das gleiche Bussystem verfügbar sind oder parallel zum PROFIBUS (als Logic-Bus) ein Motion-Bus (CANopen) benötigt wird. Die Bussysteme werden in der Steuerung synchronisiert.

Hinweis!

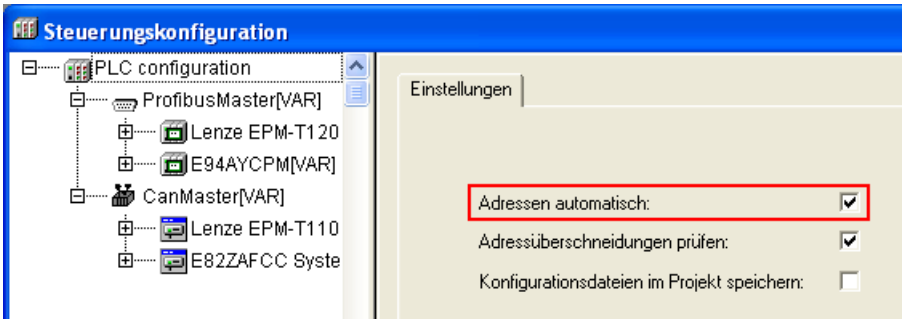
- Ein Mischbetrieb ist nur bei Industrie-PCs möglich, die zwei Erweiterungs-Schächte für Kommunikationskarten besitzen. Bei der "Command Station" ist der Mischbetrieb nicht möglich.
- Im Release 2.5 kann PROFIBUS nicht mit EtherCAT kombiniert werden.
- In der Steuerungskonfiguration muss der PROFIBUS-Master an erster Position – vor den CANopen Motion-Teilnehmern – angeordnet werden.

Kommunikationshandbuch "Steuerungstechnik CANopen"

Hier finden Sie ausführliche Informationen zur Inbetriebnahme von CANopen-Komponenten.

Adressierung der CANopen- und PROFIBUS-Teilnehmer

Die Adressvergabe für Ein- und Ausgabeobjekte der PROFIBUS- und CANopen-Teilnehmer erfolgt im »PLC Designer« automatisch (Standardeinstellung):



Hinweis!

Wir empfehlen die Standardeinstellung beizubehalten. Bei manueller Adressvergabe müssen Sie darauf achten, dass in der gesamten Steuerungskonfiguration jede Objektadresse **eindeutig** ist.

Ausführliche Informationen dazu finden Sie in der Dokumentation des »PLC Designer«.

8 Funktionsbibliotheken

Zur Konfiguration des PROFIBUS und zur Diagnose stehen im »PLC Designer« folgende Funktionsbibliotheken zur Verfügung:

Funktionsbibliotheken	Verwendung
▸ Funktionsbibliothek BusDiag.lib (📖 39)	Diese Bibliothek dient zur Abfrage von Diagnose-Informationen vom PROFIBUS-Master und von den Slaves.
▸ Funktionsbibliothek NetXPBInfo.lib (📖 44)	Diese Bibliothek dient zur Abfrage verschiedener Informationen des PROFIBUS-Masters (z. B. Fehlerzähler, Buszykluszähler).
▸ Funktionsbibliothek HilscherNetX.lib (📖 46)	Diese Bibliothek ermöglicht den direkten Zugriff auf das Paket-Interface der PROFIBUS-Kommunikationskarte.
▸ Funktionsbibliothek SysLibDPV1Hilscher.lib (📖 50)	Diese Bibliothek unterstützt die azyklischen PROFIBUS DPV1 - Klasse 1 Schreib- und Lesedienste zur Datenübermittlung zwischen dem Master und den Slaves. (Siehe dazu die PROFIBUS-DP Norm.)

Weitere Informationen zu Funktionsbibliotheken enthält die **Dokumentation/ Online-Hilfe des »PLC Designer«**.

8.1 Funktionsbibliothek BusDiag.lib

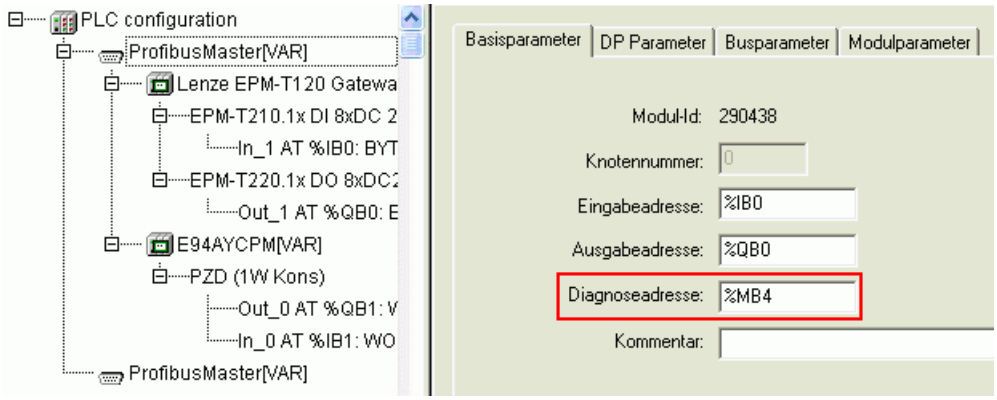
Die Funktionsbibliothek **BusDiag.lib** beinhaltet die folgenden Funktionsbausteine zur Diagnose:

- Funktionsbaustein DiagGetBusState (📖 39)
- Funktionsbaustein DiagGetState (📖 41)

8.1.1 Funktionsbaustein DiagGetBusState

Mit diesem Baustein können Sie sich den aktuellen Bus-Status anzeigen lassen.

DiagGetBusState muss mittels *AT %MByy* auf die Diagnoseadresse des zu diagnostizierenden PROFIBUS-Teilnehmers gesetzt werden. Der Status wird im Hintergrund aktualisiert. Dadurch sind die Eingangsvariablen (VAR_INPUT) nicht vorzubelegen.



Eingänge (VAR_INPUT)

Der Status wird im Hintergrund aktualisiert. Dadurch brauchen die Eingangsvariablen nicht vorbelegt zu werden.

Bezeichner/Datentyp	Bedeutung/Einstellmöglichkeiten
ENABLE BOOL	Die Aktivierung des Funktionsbausteins erfolgt flankengesteuert: • Positive Flanke (TRUE) = Diagnose-Infomationen werden ermittelt und READY wird auf TRUE gesetzt.
DRIVERNAME POINTER TO STRING	Name des Treibers (Adresse des Namens), an den der Diagnoseauftrag gehen soll. • Wird hier '0' eingetragen, wird der Diagnoseauftrag an alle vorhandenen Treiber weitergereicht.
DEVICENUMBER INT	Identifikation des Busses, der von diesem Modul (Treiber) verwaltet wird. • Die Instanz ergibt sich aus der PLC-Konfiguration. Der erste angehängte Knoten entspricht der Instanz 0, der zweite der Instanz 1 usw. • Für eine zweite Kommunikationskarte MC-PBM müssen Sie eine zweite *GetBusState*-Instanz mit einer anderen *%MByy*-Adresse erzeugen. Um die erweiterte Diagnose mit dem Funktionsbaustein DiagGetState (📖 41) zu erhalten, müssen Sie den Baustein mit *Instance := 1* aufrufen.

Ausgänge (VAR_OUTPUT)

Bezeichner/Datentyp	Bedeutung/Einstellmöglichkeiten	
READY BOOL	Immer TRUE: Die Bearbeitung des Diagnoseauftrags ist abgeschlossen.	
STATE INT	Wenn READY = TRUE, dann gibt STATE durch einen der folgenden Werte den aktuellen Status des Bausteins wieder. Diese Werte werden globalen Konstanten zugewiesen. Die Konstanten sind in der Funktionsbibliothek **BusDiag.lib** hinterlegt.	
	Wert = 1	Der Bus ist OK, kein Fehler. • Konstante BUSSTATE_BUSOK
	Wert = 2	Ein Busfehler ist aufgetreten. • Konstante BUSSTATE_BUSFAULT
	Wert = 3	Es findet keine PROFIBUS-Kommunikation statt oder die Kommunikation wurde abgebrochen. • Konstante BUSSTATE_BUSNOTCOMMUNICATING
	Wert = 4	Der PROFIBUS befindet sich im Status STOPPED. • Konstante BUSSTATE_BUSSTOPPED
EXTENDEDINFO ARRAY [0...129] OF BYTE	1:1-Relation zwischen dem ARRAY-Index und der Stationsnummer des Slaves. Nur die ersten 3 Bits des Byte werden verwendet:	
	Bit 0	Der PROFIBUS-Teilnehmer wird konfiguriert.
	Bit 1	Der PROFIBUS-Teilnehmer ist aktiv am Bus.
	Bit 2	Der PROFIBUS-Teilnehmer sendet eine Fehlermeldung. • Ausführliche Informationen können Sie über den Funktionsbaustein DiagGetState (📖 41) erhalten.

Hinweis!

Nach dem Aufstarten des Busses setzen die Slaves das Error-Flag (Bit 2), damit der Master zunächst die Diagnose-Informationen ausliest. Werden die Diagnose-Informationen für den jeweiligen Slave über den Funktionsbaustein DiagGetState (📖 41) ausgelesen, wird das Error-Flag zurückgesetzt.

8.1.2 Funktionsbaustein DiagGetState

Meldet ein auf dem Bus verfügbarer Teilnehmer einen Fehler, kann seine spezifische Diagnose-Information mit dem Baustein **DiagGetState** gelesen werden.

DiagGetState muss explizit mit der Device-Nummer und der Busmember-ID (Stationsadresse) aufgerufen werden.

Eingänge (VAR_INPUT)

Bezeichner/Datentyp	Bedeutung/Einstellmöglichkeiten
ENABLE BOOL	Die Aktivierung des Funktionsbausteins erfolgt flankengesteuert: • Positive Flanke (TRUE) = Diagnose-Infomationen werden ermittelt und READY wird auf TRUE gesetzt.
DRIVERNAME POINTER TO STRING	Immer auf '0' setzen: Der Diagnoseauftrag wird an alle vorhandenen Treiber weitergereicht.
DEVICENUMBER INT	Identifikation des Busses, der von diesem Modul (Treiber) verwaltet wird. • Die Instanz ergibt sich aus der PLC-Konfiguration. Der erste angehängte PROFIBUS-Teilnehmer entspricht der Instanz 0, der zweite der Instanz 1 usw. • Beispielsweise hat man für eine zweite Kommunikationskarte MC-PBM eine zweite *GetBusState*-Instanz an einer anderen *%MByy*-Adresse. Um die erweiterte Diagnose mit dem Funktionsbaustein DiagGetState (📖 41) zu erhalten, müssen Sie den Baustein mit *Instance := 1* aufrufen.
BUSMEMBERID DWORD	Stationsadresse des PROFIBUS-Teilnehmers, für den die Diagnose-Informationen angefordert werden sollen.

Ausgänge (VAR_OUTPUT)

Bezeichner/Datentyp	Bedeutung/Einstellmöglichkeiten	
READY BOOL	FALSE: Der Diagnoseauftrag wird bearbeitet. TRUE: Die Bearbeitung des Diagnoseauftrags ist abgeschlossen.	
STATE INT	Wenn READY = TRUE, dann gibt STATE durch einen der folgenden Werte den aktuellen Status des Bausteins wieder. Diese Werte werden globalen Konstanten zugewiesen. Die Konstanten sind in der Funktionsbibliothek **BusDiag.lib** hinterlegt.	
	Wert = -1	Ungültiger Eingangsparameter • Konstante NDSTATE_INVALID_INPUTPARAM
	Wert = 0	Diagnose-Informationen sind nicht freigegeben. • Konstante NDSTATE_NOTENABLED
	Wert = 1	Diagnose-Informationen sind angefordert. • Konstante NDSTATE_GETDIAG_INFO
	Wert = 2	Diagnose-Informationen sind verfügbar. • Konstante NDSTATE_DIAGINFO_AVAILABLE
	Wert = 3	Keine Diagnose-Informationen verfügbar (Fehler beim Lesen der Daten). • Konstante NDSTATE_DIAGINFO_NOTAVAILABLE

Bezeichner/Datentyp	Bedeutung/Einstellmöglichkeiten	
EXTENDEDINFO ARRAY [0...129] OF BYTE	Enthält die slave-spezifischen Diagnose-Informationen.	
	Byte 0	Stationsstatus 1
	Byte 1	Stationsstatus 2
	Byte 2	Stationsstatus 3
	Byte 3	Master-Stationsnummer
	Byte 4	Herstellerkennung (High Byte)
	Byte 5	Herstellerkennung (Low Byte)
	Byte 6 ... n	Slave-spezifische Diagnose-Informationen (siehe Dokumentation des Slaves)

Beispiel: Diagnose-Informationen von Servo Drives 9400 in 'EXTENDEDINFO'

Byte in EXTENDEDINFO	Bedeutung
6	Bit 0: Station exisitiert nicht (vom Master gesetzt). Bit 1: Slave ist nicht für den Datenaustausch bereit. Bit 2: Konfigurationsdaten stimmen nicht überein. Bit 3: Slave hat erweiterte Diagnosedaten. Bit 4: Angeforderte Funktion wird vom Slave nicht unterstützt. Bit 5: Ungültige Antwort vom Slave (vom Master gesetzt) Bit 6: Falsche Parametrierung Bit 7: Slave ist von einem anderen Master parametriert (vom Master gesetzt).
7	Bit 0: Slave muss neu parametriert werden. Bit 1: Statische Diagnose Bit 2: Fest auf "1" gesetzt. Bit 3: Watchdog aktiv Bit 4: Freeze-Kommando erhalten. Bit 5: Sync-Kommando erhalten. Bit 6: Reserviert Bit 7: Slave ist deaktiviert (vom Master gesetzt).
8	Bit 7: Diagnostic-Overflow - Slave hat mehr Diagnose-Informationen als in ein Telegramm passen.
9	Bits 0 ... 7: Master-Adresse nach Parametrierung ("0xFF" ohne Parametreierung)
10	Bits 0 ... 7: Identnummer High-Byte
11	Bits 0 ... 7: Identnummer Low-Byte
12	Header • Im Header wird die Blocklänge der erweiterten Diagnose inklusive des Headerbyte dargestellt. • Im vorliegenden Fall beträgt der Wert des Eintrags"0x0A" (Bytes 6 ... 15 = 10 Bytes).
13	Status_Type • Der Wert des Eintrags ist fest und beträgt "0x81" bei folgenden Bit-Belegungen: – Bit 7 = 1: "Status" – Bit 0 = 1: "Statusmeldung" – Wert aller restlichen Bits = 0
14	Slot_Number • Wert der Slot-Nummer: "0x00"
15	Specifier • Ein gemeldeter Fehler wird im Specifier mit der Kennung "0x0" (Status kommt) eingetragen. • Ein beseitigter Fehler wird im Specifier mit der Kennung "0x02" (Status geht) eingetragen. • Wenn kein Fehler gemeldet wurde, hat der Eintrag im Specifier den Wert "0x00" (keine weitere Unterscheidung).

Byte in EXTENDEDINFO	Bedeutung
16	PROFIsafe, Fehlernummer vom Sicherheitsmodul • Bei einem Fehler des Sicherheitsmoduls enthalten Byte 10 (Low-Byte) und Byte 11 (High-Byte) die entsprechende Fehlernummer. • Weitere Informationen finden Sie in der Dokumantation des entsprechenden Sicherheitsmoduls.
17	
18 ... 21	Fehlercode des PROFIBUS-Teilnehmers • Weitere Informationen finden Sie in der Dokumantation des entsprechenden PROFIBUS-Teilnehmers.

8.2 Funktionsbibliothek NetXPBInfo.lib

Mit der Funktionsbibliothek **NetXPBInfo.lib** können verschiedene Informationen zum PROFIBUS-Master abgefragt werden (z. B. Fehlerzähler, Buszykluszähler).

8.2.1 Struktur NETXGETPBINFOSTYP

```
TYPE NETXGETPBINFOSTYP :
STRUCT
   iDev : INT;
   iDummy : INT;
   ulLastCycleTime : DWORD;
   ulCycleStarts : DWORD;
   ulCycleEnds : DWORD;
   usBus_error_cnt : WORD;
   usMsg_Timeout : WORD;
   usRX_Overflow : WORD;
   usBus_Off_cnt : WORD;
   ucErrNumber : BYTE;
   ucErrStationAdr: BYTE;
END_STRUCT
END_TYPE
```

Beschreibung der Komponenten

Bezeichner/Datentyp	Bedeutung/Einstellmöglichkeiten
iDev INT	Devicenummer des PROFIBUS-Masters
iDummy INT	Nicht verwendet
ulLastCycleTime DWORD	Die letzte gemessene Zeit, um den Buszyklus zu starten.
ulCycleStarts DWORD	Anzahl der gestarteten Buszyklen
ulCycleEnds DWORD	Anzahl der abgeschlossenen Buszyklen
usBus_error_cnt WORD	Anzahl der Busfehler-Events
usMsg_Timeout WORD	Anzahl der Bus Message Fehler
usRX_Overflow WORD	Anzahl der Sende-Überlauf-Events
usBus_Off_cnt WORD	Anzahl der Bus_OFF-Events
ucErrNumber POINTER TO BYTE	Allgemeiner Fehlerzähler
ucErrStationAdr POINTER TO BYTE	Letzte fehlerhafte Slave-Adresse

8.2.2 Funktion NetXGetPBInfos

Mit dieser Funktion können verschiedene Informationen zum NetX PROFIBUS-Master abgefragt werden. Die Informationen werden in den übergebenen Speicher eingetragen. Der Rückgabewert enthält den Fehlercode. ’0’ zeigt einen Fehler an, ungleich ’0’ eine erfolgreiche Abfrage.

Eingänge (VAR_INPUT)

Bezeichner/Datentyp	Bedeutung/Einstellmöglichkeiten
pAddress POINTER TO NETXGETPBINFOSTYP	Zeiger auf eine Instanz der Struktur **NETXGETPBINFOSTYP**. In diese Struktur werden die Daten eingetragen.

8.3 Funktionsbibliothek HilscherNetX.lib

Die Funktionsbibliothek **HilscherNetX.lib** ermöglicht den direkten Zugriff auf das Paket-Interface der PROFIBUS-Kommunikationskarte.

Um die Funktionen der Bibliothek zu nutzen, ist es notwendig die entsprechende Dokumentation zur jeweiligen NetX Hilscher-Firmware zur Verfügung zu haben. Die Dokumentation können Sie über den Lenze Service beziehen.

8.3.1 Struktur CIFX_PACKET

```
TYPE CIFX_PACKET
STRUCT
   tHeader : CIFX_PACKET_HEADER;
   abData : ARRAY [0..1559] OF BYTE;
END_STRUCT
END_TYPE
```

Beschreibung der Komponenten

Bezeichner/Datentyp	Bedeutung/Einstellmöglichkeiten
tHeader POINTER TO CIFX_PACKET_HEADER	Kommando- und Verwaltungsdaten des Paketes.
abData ARRAY [0...1559] OF BYTE	Zu sendende oder empfangene Daten

8.3.2 Struktur CIFX_PACKET_HEADER

```
TYPE CIFX_PACKET_HEADER :
STRUCT
   ulDest : UDINT;
   ulSrc : UDINT;
   ulDestId : UDINT;
   ulSrcId : UDINT;
   ulLen : UDINT;
   ulId : UDINT;
   ulState : UDINT;
   ulCmd : UDINT;
   ulExt : UDINT;
   ulRout : UDINT;
END_STRUCT
END_TYPE
```

Beschreibung der Komponenten

Bezeichner/Datentyp	Bedeutung/Einstellmöglichkeiten
ulDest UDINT	Ziel des Paketes im Prozessablauf
ulSrc UDINT	Quelle des Paketes im Prozessablauf
ulDestId UDINT	Zielreferenz des Paketes
ulSrcId UDINT	Quellreferenz des Paketes
ulLen UDINT	Länge des Paketes ohne Header
ulId UDINT	Identifikations-Handle des Senders
ulState UDINT	Status des Auftrags
ulCmd UDINT	Paket-Kommando • Siehe NetX-Dokumentation für ausführliche Informationen.
ulExt UDINT	Extension • Wird nicht benutzt (Wert = ’0’)
ulRout UDINT	Routing • Wird nicht benutzt (Wert = ’0’)

8.3.3 Funktion CIFXGetChannelHandle

Diese Funktion liefert als Rückgabewert das Handle des NetX-Kanals. Dieses Handle kann verwendet werden, um die Funktionen **CIFXGetPacket** und **CIFXPutPacket** aufzurufen. Im Fehlerfall wird '0' zurück gegeben.

Eingänge (VAR_INPUT)

Bezeichner/Datentyp	Bedeutung/Einstellmöglichkeiten
iDevice INT	Device-Nummer des NetX-Knotens

8.3.4 Funktion CIFXPutPacket

Diese Funktion versendet ein Paket an den NetX-Kanal. Dazu muss ein Paket vom Typ CIFX_PACKET übergeben werden. Beim Versenden wird der Wert *ulSrcId* vom Treiber beschrieben. Dies ist wichtig, um die Antwort auf die Anfrage abzuholen. Der Rückgabewert ist der Fehlercode. Dabei ist '0' der fehlerfreie Fall und ungleich '0' der Fehlerfall. In der Hilscher Firmware-Dokumentation werden die Fehlercodes ausführlich beschrieben.

Eingänge (VAR_INPUT)

Bezeichner/Datentyp	Bedeutung/Einstellmöglichkeiten
hChannel UDINT	Handle des Kanals, das über **CIFXGetChannelHandle** erfragt werden kann.
pPacket POINTER TO CIFX_PACKET	Zeiger auf das Paket, das versendet werden soll.
ulTimeout UDINT	Versende-Timeout, das abgewartet wird.

8.3.5 Funktion CIFXGetPacket

Diese Funktion holt die Antwort auf ein zuvor versendetes Paket ab. Dazu müssen die Verwaltungsdaten des versendeten Paketes unverändert übernommen werden. Wichtig ist, dass der vom Treiber geänderte Wert *ulSrcId* übergeben wird. Der Rückgabewert ist der Fehlercode. Dabei ist '0' der fehlerfreie Fall und ungleich '0' der Fehlerfall. In der Hilscher Firmware-Dokumentation werden die Fehlercodes ausführlich beschrieben.

Eingänge (VAR_INPUT)

Bezeichner/Datentyp	Bedeutung/Einstellmöglichkeiten
hChannel UDINT	Handle des Kanals, dass über **CIFXGetChannelHandle** erfragt werden kann.
udiSize UDINT	Größe des Speicherbereichs, der mit *pPacket* übergeben wird.
pPacket POINTER TO CIFX_PACKET	Zeiger auf das Paket, das versendet werden soll.
ulTimeout UDINT	Empfangs-Timeout, das abgewartet wird.
ulRemAddress UDINT	Auf '0' gesetzt.

8.4 Funktionsbibliothek SysLibDPV1Hilscher.lib

Die Funktionsbibliothek **SysLibDPV1Hilscher.lib** unterstützt die azyklischen PROFIBUS DPV1 - Klasse 1 Schreib- und Lesedienste zur Datenübermittlung zwischen dem Master und den Slaves. Die Daten werden innerhalb der Slave-Teilnehmer über Slot und Index adressiert (siehe dazu die PROFIBUS-DP Norm).

Wenn es das Zielsystem unterstützt, sind folgende Funktionsbausteine verfügbar:

- Funktionsbaustein DPV1_Read / DPV1_ReadEx (📖 51)
- Funktionsbaustein DPV1_Write / DPV1_WriteEx (📖 52)

8.4.1 Struktur V1State

Diese Struktur wird von den Bausteinen der Funktionsbibliothek **SysLibDPV1Hilscher.lib** jeweils in der Ausgangsvariable *State* verwendet. Sie beschreibt den Status des Auftrags.

```
TYPE V1State :
   (NotEnabled := 0, InvalidParam, Started, Done, DoneWithError );
END_TYPE
```

Beschreibung der Komponenten

Bezeichner/Datentyp	Bedeutung/Einstellmöglichkeiten
NotEnabled := 0	Der Funktionsbaustein ist nicht aktiv.
InvalidParam	Ungültiger Eingangsparameter
Started	Der Funktionsbaustein hat mit der Abarbeitung begonnen.
Done	Der Funktionsbaustein hat die Abarbeitung beendet.
DoneWithError	Der Funktionsbaustein hat die Abarbeitung mit einem Fehler abgebrochen.

8.4.2 Funktionsbaustein DPV1_Read / DPV1_ReadEx

Dieser Funktionsbaustein dient zum Lesen von Daten.

Eingänge (VAR_INPUT)

Bezeichner/Datentyp	Bedeutung/Einstellmöglichkeiten
ENABLE BOOL	Die Aktivierung des Funktionsbausteins erfolgt flankengesteuert: • Positive Flanke (TRUE) = Diagnose-Infomationen werden ermittelt und READY wird auf TRUE gesetzt.
Device INT	Index der Hilscher-Karte, an die der Auftrag übergeben wird.
StationAddr INT	Stationsadresse des Slaves im PROFIBUS.
Slot INT	Daten-Slot, zur Identifizierung der Daten im Slave.
Index INT	Daten-Index, zur Identifizierung der Daten im Slave.
Len INT	Länge der zu lesenden/schreibenden Daten in Bytes. • Hier wird die maximale Länge des Datenpuffers *buffer* übergeben.
buffer DWORD	Die lokale Adresse der Daten (mit ADR() bilden).

Ausgänge (VAR_OUTPUT)

Bezeichner/Datentyp		Bedeutung/Einstellmöglichkeiten
Ready BOOL		FALSE: Der Leseauftrag wird bearbeitet. TRUE: Der Leseauftrag ist abgeschlossen.
State POINTER TO V1State		Auskunft über den Zustand des Auftrags • siehe Struktur V1State (📖 50)
Size INT		Länge der tatsächlich gelesenen/geschriebenen Daten bei erfolgreicher Ausführung.
Error ARRAY [0 ... 7] OF BYTE		Informationen zu aufgetretenen Fehlern • Nur in Funktionsbaustein **DPV1_ReadEx** enthalten.
	Byte 1	Hilscher Fehlercode • Siehe Dokumentation zu Hilscher PROFIBUS-Karten "Protocol Interface Manual Profibus DP" für ausführliche Informationen.
	Byte 2	Fehlerklassen-Code • Siehe PROFIBUS-Norm für ausführliche Informationen.
	Byte 3 + 4	Index 2 + 3: Slave-spezifische Fehlerinformation • Siehe Dokumentation des Slaves für ausführliche Informationen.

8.4.3 Funktionsbaustein DPV1_Write / DPV1_WriteEx

Dieser Funktionsbaustein dient zum Schreiben von Daten.

Eingänge (VAR_INPUT)

Bezeichner/Datentyp	Bedeutung/Einstellmöglichkeiten
ENABLE BOOL	Die Aktivierung des Funktionsbausteins erfolgt flankengesteuert: • Positive Flanke (TRUE) = Diagnose-Infomationen werden ermittelt und READY wird auf TRUE gesetzt.
Device INT	Index der Hilscher-Karte, an die der Auftrag übergeben wird.
StationAddr INT	Stationsadresse des Slaves im PROFIBUS.
Slot INT	Daten-Slot, zur Identifizierung der Daten im Slave.
Index INT	Daten-Index, zur Identifizierung der Daten im Slave.
Len INT	Länge der zu lesenden/schreibenden Daten in Bytes. • Hier wird die maximale Länge des Datenpuffers *buffer* übergeben.
buffer DWORD	Die lokale Adresse der Daten (mit ADR() bilden).

Ausgänge (VAR_OUTPUT)

Bezeichner/Datentyp		Bedeutung/Einstellmöglichkeiten
Ready BOOL		FALSE: Der Leseauftrag wird bearbeitet. TRUE: Der Leseauftrag ist abgeschlossen.
State POINTER TO V1State		Auskunft über den Zustand des Auftrags • siehe Struktur V1State (📖 50)
Size INT		Länge der tatsächlich gelesenen/geschriebenen Daten bei erfolgreicher Ausführung.
Error ARRAY [0 ... 7] OF BYTE		Informationen zu aufgetretenen Fehlern • Nur in Funktionsbaustein **DPV1_WriteEx** enthalten.
	Byte 1	Hilscher Fehlercode • Siehe Dokumentation zu Hilscher PROFIBUS-Karten "Protocol Interface Manual Profibus DP" für ausführliche Informationen.
	Byte 2	Fehlerklassen-Code • Siehe PROFIBUS-Norm für ausführliche Informationen.
	Byte 3 + 4	Index 2 + 3: Slave-spezifische Fehlerinformation • Siehe Dokumentation des Slaves für ausführliche Informationen.

8.4.4 Telegramm-Beispiele für den PROFIdrive-Parameterdatenkanal (DP-V1)

Im Folgenden wird ein Parameterleseauftrag und ein Parameterschreibauftrag bei einem Servo Drive 9400 beschrieben.

8.4.4.1 Beispiel Leseauftrag: Kühlkörpertemperatur abfragen

Die Kühlkörpertemperatur des Servo Drive 9400 soll gelesen werden.

- Zu lesende Codestelle: C00061
- Kühlkörpertemperatur: 43 °C

Parameterauftrag

Byte 1	Byte 2	Byte 3	Byte 4
Auftragsreferenz	Auftragskennung	Achse	Anzahl Indizes
0xXX	0x01	0x00	0x01
	Parameter zum Lesen anfordern		

Byte 5	Byte 6
Attribut	Anzahl Subindizes
0x10	0x00
Wert	Kein Subindex

Byte 7	Byte 8	Byte 9	Byte 10
Index		Subindex	
High-Byte	Low-Byte	High-Byte	Low-Byte
0x5F	0xC2	0x00	0x00
Index = 24575 - Codestellen-Nr. = 24575 - 61 = 24514 = 0x5F C2		Kein Subindex	

Parameterantwort nach fehlerfreien Leseauftrag

Byte 1	Byte 2	Byte 3	Byte 4
Auftragsreferenz	Antwortkennung	Achse	Anzahl Indizes
0xXX	0x01	0x00	0x01
(gespiegelt)	Parameter gelesen	(gespiegelt)	

Byte 5	Byte 6
Format	Anzahl Werte
0x43	0x01
Doppelwort	1 Wert

Byte 7	Byte 8	Byte 9	Byte 10
Wert			
High-Wort: High-Byte	High-Wort: Low-Byte	Low-Wort: High-Byte	Low-Wort: Low-Byte
0x00	0x00	0x00	0x2B
Gelesener Wert = 0x00 00 00 2B = 43 x 1 (interner Faktor) = 43 [°C]			

Parameterantwort nach Lesefehler

Byte 1	Byte 2	Byte 3	Byte 4
Auftragsreferenz	Antwortkennung	Achse	Anzahl Indizes
0xXX	0x81	0x00	0x01
(gespiegelt)	Parameter nicht gelesen	(gespiegelt)	

Byte 5	Byte 6
Format	Anzahl Werte
0x44	0x01
Fehler	Fehlercode ohne Zusatzinformation

Byte 7	Byte 8
Fehlercode	
High-Byte	Low-Byte
Siehe Dokumentation des PROFIBUS-Teilnehmers.	

8.4.4.2 Beispiel Schreibauftrag: Ablaufzeit für Schnellhalt einstellen

Im Servo Drive 9400 soll die Ablaufzeit für Schnellhalt auf 50 ms eingestellt werden.

▶ Zu beschreibende Codestelle: C00105

Parameterauftrag

Byte 1	Byte 2	Byte 3	Byte 4
Auftragsreferenz	Auftragskennung	Achse	Anzahl Indizes
0xXX	0x02	0x00	0x01
	Parameter schreiben	Achse 0	1 Index

Byte 5	Byte 6
Attribut	Anzahl Subindizes
0x10	0x00
Wert	Kein Subindex

Byte 7	Byte 8	Byte 9	Byte 10
Index		Subindex	
High-Byte	Low-Byte	High-Byte	Low-Byte
0x5F	0x96	0x00	0x00
Index = 24575 - Codestellen-Nr. = 24575 - 105 = 24470 = 0x5F 96		Kein Subindex	

Byte 11	Byte 12
Format	Anzahl Werte
0x43	0x01
Doppelwort	1 Wert

Byte 13	Byte 14	Byte 15	Byte 16
Wert			
High-Wort: High-Byte	High-Wort: Low-Byte	Low-Wort: High-Byte	Low-Wort: Low-Byte
0x00	0x00	0x00	0x32
Zu schreibender Wert = 0,05 [s] x 1000 (interner Faktor) = 50 = 0x00 00 00 32			

Parameterantwort nach fehlerfreien Schreibauftrag

Byte 1	Byte 2	Byte 3	Byte 4
Auftragsreferenz	Antwortkennung	Achse	Anzahl Indizes
0xXX	0x02	0x00	0x01
(gespiegelt)	Parameter beschrieben	(gespiegelt)	1 Index

Parameterantwort nach Lesefehler

Byte 1	Byte 2	Byte 3	Byte 4
Auftragsreferenz	Antwortkennung	Achse	Anzahl Indizes
0xXX	0x82	0x00	0x01
(gespiegelt)	Parameter nicht beschrieben	(gespiegelt)	1 Index

Byte 5	Byte 6
Format	Anzahl Werte
0x44	0x01
Fehler	Fehlercode ohne Zusatzinformation

Byte 7	Byte 8
Fehlercode	
High-Byte	Low-Byte
Siehe Dokumentation des PROFIBUS-Teilnehmers.	

9 Minimale Zykluszeit des PLC-Projektes bestimmen

In diesem Kapitel erfahren Sie, wie Sie die minimale Zykluszeit des PLC-Projektes bestimmen können.

Die Berechnung der minimalen Zykluszeit gliedert sich in folgende Schritte:

1. Gesamtzugriffszeit $T_{Korrektur}$ auf die Peripheriegeräte berechnen.
   ▸ Gesamtzugriffszeit auf die Peripheriegeräte berechnen ($T_{Korrektur}$) (📖 57)
2. Die Taskauslastung $T_{Taskauslastung}$ der Applikation während des Betriebs ermitteln.
   ▸ Taskauslastung der Applikation ermitteln ($T_{Taskauslastung}$) (📖 58)
3. Die minimale Zykluszeit berechnen.
   ▸ Minimale Zykluszeit berechnen (📖 60)
4. System optimieren.
   ▸ System optimieren (📖 61)

9.1 Gesamtzugriffszeit auf die Peripheriegeräte berechnen ($T_{Korrektur}$)

Die Zugriffszeiten sind abhängig von der Anzahl der konfigurierten Feldgeräte und von der eingesetzten IPC-Hardware:

Konfiguration	Zugriffszeit mit Prozessor ATOM 1.6 GHz
1 PROFIBUS-Master (Logic)	140 µs
1. Achse (16 Eingangsworte und 16 Ausgangsworte)	40 µs
je weitere Achse (16 Eingangsworte und 16 Ausgangsworte)	10 µs

Beispiel

Zugriffszeiten Industrie-PC (ATOM 1.6 GHz) mit 3 Achsen	
Zugriffszeit PROFIBUS-Master	140 µs
+ Zugriffszeit 1. Achse	40 µs
+ Zugriffszeit 2 weitere Achsen	20 µs
= Gesamtzugriffszeit	200 µs

9.2 Taskauslastung der Applikation ermitteln ($T_{Taskauslastung}$)

Die Zeit $T_{Taskauslastung}$ kann nicht errechnet werden. Sie wird am laufenden System ermittelt. Dazu wird das System mit einer ausreichend großen Zykluszeit in Betrieb genommen und anschließend optimiert.

Um die Taskauslastung zu ermitteln, verwenden Sie den Task-Editor im »PLC Designer«.

9.2.1 Anzeige der Systemauslastung im »PLC Designer« mit dem Task-Editor

Hinweis!

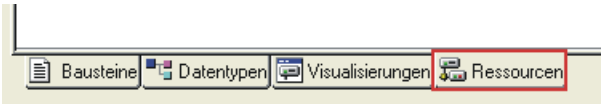
Um die Auslastung für alle Tasks anzeigen zu können, muss die IEC 61131-Bibliothek *SysTaskInfo* im Projekt eingefügt sein.

Der Task-Editor enthält ein zweigeteiltes Dialogfenster:

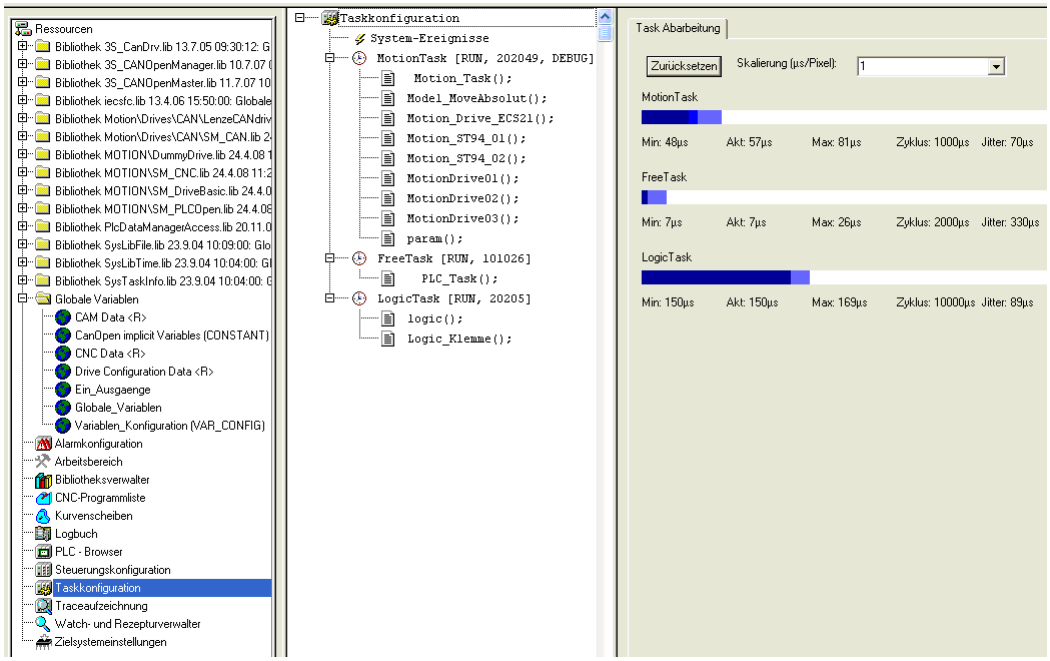
- Auf der linken Seite sehen Sie die Tasks in einem Konfigurationsbaum.
- Ist der Eintrag *Taskkonfiguration* markiert, wird die Auslastung für alle Tasks in Balkendiagrammen im rechten Dialogfenster angezeigt.

So zeigen Sie die Systemauslastung an:

1. Registerkarte *Ressourcen* auswählen:



2. *Taskkonfiguration* im Online-Modus des »PLC Designer« öffnen:



9.2.2 Taskauslastung ermitteln

Ausgangssituation

Ein Projekt mit z. B. einer Motiontask und 2 niederprioren Tasks ist vollständig erstellt.

So ermitteln Sie die Taskauslastung $T_{Taskauslastung}$:

1. Für eine erste Messung von $T_{Taskauslastung}$ werden die Zykluszeiten aller im PLC-System vorhandenen zyklischen Tasks "groß" eingestellt.
   - Beispiel: Motiontask = 10 ms, alle anderen zyklischen Tasks = 20 ms
2. Einloggen und Projekt laden.
3. Nach dem vollständigen Hochlauf des Systems die Schaltfläche **Zurücksetzen** auf der Registerkarte *Task Abarbeitung* betätigen.
   - Die angezeigten Tasklaufzeiten werden zurückgesetzt.
4. Die in der Taskkonfiguration angezeigte maximale Rechenzeit der höchstprioren Task ablesen = $T_{Taskauslastung}$.

9.3 Minimale Zykluszeit berechnen

Hinweis!

Zur Berechnung der minimalen Zykluszeit wird ein Sicherheitsfaktor von 1.5 einbezogen.

Die minimale Zykluszeit T_{min} für ein System ergibt sich aus der Summe der zuvor ermittelten Zeiten, multipliziert mit dem Sicherheitsfaktor:

$$T_{min} > \text{Sicherheitsfaktor} \times (T_{Taskauslastung} + T_{Korrektur})$$

Beispiel

Konfiguration: Industrie-PC (ATOM 1.6 GHz) mit 3 Achsen		
Ermittelte Zugriffszeit		**Ergebnis**
Errechneter Korrekturwert	$T_{Korrektur}$	**200 µs** (140 µs + 40 µs +(2 x 10 µs))
Abgelesener Wert aus Taskkonfiguration:	$T_{Taskauslastung}$	**500 µs**
Tatsächlicher Rechenzeitbedarf		**700 µs**
Minimale Zykluszeit inklusive Sicherheitsfaktor 1.5	T_{min}	**1050 µs**
Tatsächlich gewählte Zykluszeit		**2000 µs**

9.4 System optimieren

So optimieren Sie das System:

1. Einloggen und Projekt laden.
2. Task-Abarbeitungszeiten kontrollieren.
3. Zykluszeiten optimieren:
   - Falls technologisch notwendig können die Zykluszeiten der restlichen niederprioren Tasks verkürzt werden.
   - Bedingung: Keine niederpriore Task darf in ihrer Taskauslastung mehr als 60 % der jeweiligen Zykluszeit belegen.

10 Diagnose

Die Diagnose des PROFIBUS können Sie im »PLC Designer« mit Hilfe von bestimmten Diagnose-Funktionsbausteinen durchführen.

Zudem verfügen die Feldgeräte, PROFIBUS-Kommmunikationsmodule und die Kommunikationskarte MC-PBM über LED-Statusanzeigen zur Diagnose.

10.1 Diagnose im »PLC Designer«

Für die Diagnose des PROFIBUS im »PLC Designer« stellt die Funktionsbibliothek BusDiag.lib (📖 39) zwei Funktionsbausteine bereit:

- Funktionsbaustein DiagGetBusState (📖 39):
  - Gibt den aktuellen Bus-Status wieder.
- Funktionsbaustein DiagGetState (📖 41)
  - Meldet ein auf dem Bus verfügbarer Teilnehmer einen Fehler, kann seine spezifische Diagnose-Information gelesen werden.

10.2 Fehlermeldungen bei nicht vorhandener Kommunikationskarte MC-PBM

Ist keine Kommunikationskarte MC-PBM in den Industrie-PC eingebaut, kommt es beim Download des »PLC Designer«-Projektes zu Fehlermeldungen.

Abhilfe: Kommunikationskarte MC-PBM in den Industrie-PC einbauen.

10.3 LED-Statusanzeigen

Die LED-Statusanzeigen der Feldgeräte, PROFIBUS-Kommmunikationsmodule und der Kommunikationskarte MC-PBM geben Informationen zu Geräte- und Kommunikationszuständen wieder.

Ausführliche Informationen zu den LED-Statusanzeigen der Feldgeräte und PROFIBUS-Kommmunikationsmodule finden Sie in den entsprechenden Dokumentationen.

LED-Statusanzeigen der Kommunikationskarte MC-PBM

LED	Farbe	Zustand	Beschreibung
SYS	grün	Ein	Kommunikation aktiv: Das Gerät hat mindestens eine Verbindung zu einem konfigurierten Teilnehmer aufgebaut.
		Blinkt 5-mal pro Sekunde (5 Hz)	Kein Fehler in der Konfiguration: Die Kommunikation ist gestoppt oder das Gerät ist bereit für die Kommunikation. Es besteht aber keine Verbindung zu einem Slave.
		Blinkt unregelmäßig	• Anlauf: Fehlende oder fehlerhafte Konfiguration • Laufzeit: Host-Watchdog Zeitfehler
	gelb	Blinkt 1-mal pro Sekunde (1 Hz)	Das Gerät ist im Bootstraploader-Modus und wartet auf das Firmware-Download
		Blinkt 5-mal pro Sekunde (5 Hz)	Das Firmware-Download wird durchgeführt.
		Blinkt unregelmäßig	Ein Hardware- oder schwerer Systemfehler wurde erkannt.
	-	Aus	Keine Spannungsversorgung oder die Hardware ist defekt.
ST0	-	Aus	Keine Funktion
ST1	-	Aus	Keine Funktion
ST2	rot	Ein	Das Gerät hat ein Kommunikationsproblem zu mindestens einem PROFIBUS-Slave oder hat einen Kurzschluss erkannt.
	gelb	Ein	Das Gerät hält den PROFIBUS-Token und kann Telegramme übertragen.
		Blinkt unregelmäßig	Das Gerät ist im PROFIBUS-Netzwerk und teilt das Token mit anderen PROFIBUS Master-Geräten.
	-	Aus	Eine Verbindung zum PROFIBUS besteht.

11 Parameter-Referenz

Dieses Kapitel ergänzt die Parameterliste der Online-Dokumentation zum Industrie-PC um die Parameter der **Kommunikationskarte MC-PBM**:

▸ Parameter der Kommunikationskarte MC-PBM in Steckplatz 1 (📖 65)
▸ Parameter der Kommunikationskarte MC-PBM in Steckplatz 2 (📖 66)

Hinweis!

- Pro Industrie-PC sind mehrere Kommunikationskarten MC-PBM verwendbar. Die Bezeichnung der Karte in der »WebConfig« ist *MC-PBM*.
- Je nach verwendetem Schacht unterscheiden sich die Codestellenummern um einen Offset von ’500’. So ist für eine Kommunikationskarte in Schacht 2 ein Offset von ’500’ zu den Codestellenummern einer Karte in Schacht 1 addiert.

Tipp!

Allgemeine Informationen zu Parametern finden Sie in der Online-Dokumentation zum Industrie-PC.

11.1 Parameter der Kommunikationskarte MC-PBM in Steckplatz 1

Die Parameter sind in numerisch aufsteigender Reihenfolge aufgeführt.

C1031

Parameter | Name:	Datentyp: VISIBLE_STRING
C1031 | Gerät: Kennung	Index: $23544_d = 5BF8_h$

Identifikation der Karte

☑ Lesezugriff ☐ Schreibzugriff ☐ RSP ☐ PLC-STOP ☐ Kein Transfer

C1032

Parameter | Name:	Datentyp: VISIBLE_STRING
C1032 | Gerät: Version	Index: $23543_d = 5BF7_h$

Versionsnummer der Karte

☑ Lesezugriff ☐ Schreibzugriff ☐ RSP ☐ PLC-STOP ☐ Kein Transfer

C1033

Parameter | Name:	Datentyp: VISIBLE_STRING
C1033 | Gerät: Name	Index: $23542_d = 5BF6_h$

Gerätename der Karte

☑ Lesezugriff ☐ Schreibzugriff ☐ RSP ☐ PLC-STOP ☐ Kein Transfer

C1034

Parameter | Name:	Datentyp: VISIBLE_STRING
C1034 | Gerät: Softwareversion	Index: $23541_d = 5BF5_h$

Softwareversion der Karte

☑ Lesezugriff ☐ Schreibzugriff ☐ RSP ☐ PLC-STOP ☐ Kein Transfer

C1035

Parameter | Name:	Datentyp: VISIBLE_STRING
C1035 | Gerät: Hardwareversion	Index: $23540_d = 5BF4_h$

Hardwareversion der Karte

☑ Lesezugriff ☐ Schreibzugriff ☐ RSP ☐ PLC-STOP ☐ Kein Transfer

C1036

Parameter | Name:	Datentyp: VISIBLE_STRING
C1036 | Gerät: Seriennummer	Index: $23539_d = 5BF3_h$

Seriennummer der Karte

☑ Lesezugriff ☐ Schreibzugriff ☐ RSP ☐ PLC-STOP ☐ Kein Transfer

C1037

Parameter | Name:	Datentyp: VISIBLE_STRING
C1037 | Gerät: Hersteller	Index: $23538_d = 5BF2_h$

Hersteller der Karte

☑ Lesezugriff ☐ Schreibzugriff ☐ RSP ☐ PLC-STOP ☐ Kein Transfer

C1038

Parameter | Name:	Datentyp: VISIBLE_STRING
C1038 | Gerät: Herstelldatum	Index: $23537_d = 5BF1_h$

Herstelldatum der Karte

☑ Lesezugriff ☐ Schreibzugriff ☐ RSP ☐ PLC-STOP ☐ Kein Transfer

11.2 Parameter der Kommunikationskarte MC-PBM in Steckplatz 2

Die Parameter sind in numerisch aufsteigender Reihenfolge aufgeführt.

C1531

Parameter | Name:
C1531 | Gerät: Kennung

Datentyp: VISIBLE_STRING
Index: $23044_d = 5A04_h$

Identifikation der Karte

☑ Lesezugriff ☐ Schreibzugriff ☐ RSP ☐ PLC-STOP ☐ Kein Transfer

C1532

Parameter | Name:
C1532 | Gerät: Version

Datentyp: VISIBLE_STRING
Index: $23043_d = 5A03_h$

Versionsnummer der Karte

☑ Lesezugriff ☐ Schreibzugriff ☐ RSP ☐ PLC-STOP ☐ Kein Transfer

C1533

Parameter | Name:
C1533 | Gerät: Name

Datentyp: VISIBLE_STRING
Index: $23042_d = 5A02_h$

Gerätename der Karte

☑ Lesezugriff ☐ Schreibzugriff ☐ RSP ☐ PLC-STOP ☐ Kein Transfer

C1534

Parameter | Name:
C1534 | Gerät: Softwareversion

Datentyp: VISIBLE_STRING
Index: $23041_d = 5A01_h$

Softwareversion der Karte

☑ Lesezugriff ☐ Schreibzugriff ☐ RSP ☐ PLC-STOP ☐ Kein Transfer

C1535

Parameter | Name:
C1535 | Gerät: Hardwareversion

Datentyp: VISIBLE_STRING
Index: $23040_d = 5A00_h$

Hardwareversion der Karte

☑ Lesezugriff ☐ Schreibzugriff ☐ RSP ☐ PLC-STOP ☐ Kein Transfer

C1536

Parameter | Name:
C1536 | Gerät: Seriennummer

Datentyp: VISIBLE_STRING
Index: $23039_d = 59FF_h$

Seriennummer der Karte

☑ Lesezugriff ☐ Schreibzugriff ☐ RSP ☐ PLC-STOP ☐ Kein Transfer

C1537

Parameter | Name:
C1537 | Gerät: Hersteller

Datentyp: VISIBLE_STRING
Index: $23038_d = 59FE_h$

Hersteller der Karte

☑ Lesezugriff ☐ Schreibzugriff ☐ RSP ☐ PLC-STOP ☐ Kein Transfer

C1538

Parameter | Name:
C1538 | Gerät: Herstelldatum

Datentyp: VISIBLE_STRING
Index: $23037_d = 59FD_h$

Herstelldatum der Karte

☑ Lesezugriff ☐ Schreibzugriff ☐ RSP ☐ PLC-STOP ☐ Kein Transfer

12 Index

A

Adressierung der CANopen- und PROFIBUS-Teilnehmer 37
Anschluss PROFIBUS (SUB-D, 9-pol. Stecker) 20
Anwendungshinweise (Darstellung) 10
Anzeige der Systemauslastung im »PLC Designer« mit dem Task-Editor 58
Aufbau des PROFIBUS-Systems 15

B

Begriffe 9
Bus-Abschlusswiderstand 23
BusDiag.lib 39
Buskabellänge 21

C

C1031 | Gerät - Kennung 65
C1032 | Gerät - Version 65
C1033 | Gerät - Name 65
C1034 | Gerät - Softwareversion 65
C1035 | Gerät - Hardwareversion 65
C1036 | Gerät - Seriennummer 65
C1037 | Gerät - Hersteller 65
C1038 | Gerät - Herstelldatum 65
C1531 | Gerät - Kennung 66
C1532 | Gerät - Version 66
C1533 | Gerät - Name 66
C1534 | Gerät - Softwareversion 66
C1535 | Gerät - Hardwareversion 66
C1536 | Gerät - Seriennummer 66
C1537 | Gerät - Hersteller 66
C1538 | Gerät - Herstelldatum 66
CANopen mit PROFIBUS 37
CIFX_PACKET 46
CIFX_PACKET_HEADER 47
CIFXGetChannelHandle 48
CIFXGetPacket 49
CIFXPutPacket 48
Codestellen 64
Copyright 2

D

DiagGetBusState 39
DiagGetState 41
Diagnose 62
Diagnose im »PLC Designer« 62
Diagnose-Informationen von Servo Drives 9400 (Beispiel) 42
Dokumenthistorie 7
DPV1_Read / DPV1_ReadEx 51
DPV1_Write / DPV1_WriteEx 52

E

Ein- und Ausgabeobjekte konfigurieren 34
Engineering-Werkzeuge 22

F

Feldgeräte 18
Feldgeräte in Betrieb nehmen 25
Feldgeräte installieren 23
Funktion CIFXGetChannelHandle 48
Funktion CIFXGetPacket 49
Funktion CIFXPutPacket 48
Funktion NetXGetPBInfos 45
Funktionsbaustein DiagGetBusState 39
Funktionsbaustein DiagGetState 41
Funktionsbaustein DPV1_Read / DPV1_ReadEx 51
Funktionsbaustein DPV1_Write / DPV1_WriteEx 52
Funktionsbibliothek BusDiag.lib 39
Funktionsbibliothek HilscherNetX.lib 46
Funktionsbibliothek NetXPBInfo.lib 44
Funktionsbibliothek SysLibDPV1Hilscher.lib 50

G

Gerät
- Hardwareversion (C1035) 65
- Hardwareversion (C1535) 66
- Herstelldatum (C1038) 65
- Herstelldatum (C1538) 66
- Hersteller (C1037) 65
- Hersteller (C1537) 66
- Kennung (C1031) 65
- Kennung (C1531) 66
- Name (C1033) 65
- Name (C1533) 66
- Seriennummer (C1036) 65
- Seriennummer (C1536) 66
- Softwareversion (C1034) 65
- Softwareversion (C1534) 66
- Version (C1032) 65
- Version (C1532) 66

Gerätestammdatendateien (GSD) ablegen 24
Gesamtzugriffszeit auf die Peripheriegeräte berechnen 57

H

HilscherNetX.lib 46

I

Inbetriebnahme des PROFIBUS 22

K
Kabelspezifikation Buskabel 21
Kombination mit anderen Bussystemen 18
Kommunikationskarte MC-PBM 19
Kommunikationsmedium 20
Kurzbeschreibung PROFIBUS 14

L
LED-Statusanzeigen 63

M
Minimale Zykluszeit berechnen 60
Minimale Zykluszeit des PLC-Projektes bestimmen 57
Mischbetrieb (CANopen mit PROFIBUS) 37
Mit »PLC Designer« in die Steuerung einloggen 36

N
NetXGetPBInfos 45
NETXGETPBINFOSTYP 44
NetXPBInfo.lib 44
Netzwerktopologie 20

P
Parameter der Kommunikationskarte MC-PBM in Schacht 1 65
Parameter der Kommunikationskarte MC-PBM in Schacht 2 66
Parameter-Referenz 64
PLC-Programm anlegen 26
PLC-Programm laden und starten 36
PROFIBUS 14
PROFIBUS mit CANopen 37
PROFIBUS-Hardware für den Industrie-PC 19
PROFIBUS-Master konfigurieren 29
PROFIBUS-Slave konfigurieren 33
PROFIBUS-Verdrahtung 17
Projektdaten übersetzen 36
Projektordner anlegen 24
Protokoll 20

R
Repeater-Einsatz 17

S
Sicherheit 11
Sicherheitshinweise (Darstellung) 10
Struktur CIFX_PACKET 46
Struktur CIFX_PACKET_HEADER 47
Struktur NETXGETPBINFOSTYP 44
Struktur V1State 50
Symbolische Namen 35
SysLibDPV1Hilscher.lib 50
System optimieren 61

T
Taskauslastung der Applikation ermitteln 58
Taskauslastung ermitteln 59
Technische Daten 20
Technische Daten der Kommunikationskarte MC-PBM 20
Teilnehmer pro Netz 20
Teilnehmer pro Segment 20

U
Übertragungsrate 21

V
V1State 50
Verwendete Hinweise 10
Verwendete Konventionen 8

Z
Zielgruppe 7

© 07/2011

Lenze Automation GmbH
Grünstraße 36
D-40667 Meerbusch
Germany

+49 (0)21 32 / 99 04-0

+49 (0)21 32 / 7 21 90

Lenze@Lenze.de

www.Lenze.com

Service Lenze Service GmbH
Breslauer Straße 3
D-32699 Extertal
Germany

00 80 00 / 24 4 68 77 (24 h helpline)

+49 (0)51 54 / 82-11 12

Service@Lenze.de

KHBPBPCBAUTO ▪ 13383677 ▪ DE 2.2 ▪ TD17

10 9 8 7 6 5 4 3 2 1